VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Přehled soft clusteringovych metod, jejich aplikace nad vybranou datovou kolekcí a srovnání výsledků

Overview of soft clustering methods, their applications and comparison

2012 Bc. Jakub Měchura

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Jakub Měchura**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Přehled soft clusteringovych metod, jejich aplikace nad vybranou datovou kolekcí a srovnání výsledků
Overview of Soft Clustering Methods, their Applications and Comparison

Zásady pro vypracování:

Cílem diplomové práce je získat přehled v oblasti soft clusteringových metod, implementovat dvě různé metody a pomocí nich analyzovat vlastní datovou kolekci.

1. Nastudovat problematiku soft shlukovacích metod a vypracovat jejich přehled.
2. Získat datovou kolekci, nad kterou se budou provádět experimenty se shlukovacími algoritmy.
3. Vlastní implementace dvou různých soft clusteringových algoritmů.
4. Vizualizovat vysledky, porovnat a zhodnotit kvalitu získaných shluků.

Seznam doporučené odborné literatury:

Sadaaki Miyamoto, Hidetomo Ichihashi, Katsuhiro Honda:Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications (Studies in Fuzziness and Soft Computing)
C. Jianbin : A Graph Partition-Based Soft Clustering Algorithm
K. Yu :Soft clustering on graphs

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Mgr. Pavla Dráždilová**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava.*

V Ostravě 3. května 2012 .....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 3. května 2012 .....

Na tomto místě bych rád poděkoval vedoucí své diplomové práce paní Mgr. Pavle Dráždilové za konzultace, pomoc a cenné rady při vytváření této práce.

Abstrakt

Tématem této diplomové práce je soft clustering nebo-li soft shlukování. Cílem je vytvoření přehledu složeného z několika vybraných soft shlukovacích metod. Na základě tohoto přehledu jsou pak některé soft shlukovací metody implementovány a aplikovány nad reálnou kolekci dat, získanou prostřednictvím web scrapingu ze zvoleného diskuzního fóra. Následně jsou prováděny různé experimenty změnou nastavení vlastností algoritmů. Tyto výsledky jsou následně vizualizovány a vyhodnoceny.

Klíčová slova: soft clustering, web scraping, diskuzní fórum, fuzzy c-means, rough c-means, k-means, rough fuzzy c-means

Abstract

The theme of this thesis is soft clustering. The aim is create overview consisting of a few chosen soft clustering method. Based on this overview are chosen soft clustering methods. These methods are implemented and applied over chosen real data collection, obtained by web scraping from some discussion forum. Then, various experiments are carried out by changing the settings properties of algorithms. These results are then visualized an evaluated.

Keywords: soft clustering, web scraping, discussion forum, fuzzy c-means, rough c-means, k-means, rough fuzzy c-means

Seznam použitých zkratek a symbolů

HTML	–	Hyper Text Markup Language
RST	–	Rough Set Theory
URL	–	Uniform Resource Locator
ERD	–	Entity Relationship diagram
GPSC	–	Graph Partitioning-based Soft Clustering
PoBOC	–	Poles Based Overlappnig Clustering
FCM	–	Fuzzy c-means
RCM	–	Rough c-means
RFCM	–	Rough Fuzzy c-means

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Tématem této diplomové práce je vytvořit přehled soft shlukovacích metod a aplikovat některé z nich nad reálnou datovou kolekci. V souvislosti se shlukovacími metodami bývá častěji zmiňován pojem překrývající se shlukovací metody. Shlukovací metody umožňuji seskupovat objekty do stejných skupin, tedy shluků, na základě jejich podobnosti. Soft shlukovací metody na rozdíl od klasických shlukovacích metod umožňují, aby daný objekt patřil k více shlukům. Tím pak dochází k jejich překrývání. Na základě toho lze pak sledovat některé vlastnosti, které klasické shlukovací metody neumožňují.

Diplomová práce je rozdělena do několika vzájemně na sebe navazujících kapitol, z nichž první kapitola se zabývá popisem extrakce dat z html stránek. Jako zdroj dat bylo použito diskuzního matematického fóra, sloužící jako reálná kolekce dat, nad kterou jsou aplikovány vybrané soft shlukovací algoritmy.

Diskuzní fórum bylo použito z toho důvodu, že jej navštěvuje velké množství lidí s podobnými zájmy. Lze v něm pozorovat komunikaci uživatelů v dlouhém časovém období, z níž lze poté vysledovat různé vztahy mezi jednotlivými uživateli, např. zda jsou uživatelé tématicky vyhraněni, v jakou dobu přispívají do diskuzí apod. Soft shlukovací metody umožňují sledování právě těchto vztahů. U sociálních sítí pak můžeme pozorovat, jestli uživatelé patří do více skupin apod. Tady všude může docházet k překryvům vlivem toho, že daný objekt může spadat do více skupin (shluků).

Následující kapitolou je seznámení s problematikou soft shlukování a samotný přehled soft shlukovacích metod, kde popisujeme několik z mnoha existujících soft shlukovacích metod včetně slovního popisu algoritmů.

Následující kapitoly týkající se implementace, kde je popsán postup při implementaci vybraných soft shlukovacích algoritmů v programovacím jazyce C#, v .NET framework 4.0 a kapitola týkající se experimentů a vizualizace jejich výsledků.

Poslední kapitolou je závěr, obsahující shrnutí všech postupů a dosažených výsledků této diplomové práce.

2 Extrakce dat pro práci se soft shlukovací algoritmy

K tomu, aby se daly použít a aplikovat shlukovací algoritmy, bylo třeba získat množinu dat, nad kterou budeme dané algoritmy aplikovat. V našem případě budeme pracovat s reálnou množinou dat. Jako zdroj těchto dat se nabízí použít nějaké diskuzní fórum nebo sociální síť, jež navštěvuje velké množství lidí komunikující mezi sebou na základě společných zájmů.

V této diplomové práci jsme použili oblíbeného diskuzního matematického fóra, které sdružuje uživatele se zájmem o matematiku, fyziku apod. Jedná se o jednorázovou nebo opakovanou pomoc při řešení problémů a příkladů, popř. rady ostatním uživatelům. Oproti různým chatům a komunikačním softwarům jako je např. icq, qip, msn atd. nemusí být reakce na nové příspěvky okamžitá. Uživatelé na ně mohou reagovat s určitým časovým odstupem. Nicméně pro aplikaci daných shlukovacích algoritmů je toto diskuzní matematické fórum ideální, jelikož nám získaná data umožní sledovat různé závislosti a vztahy mezi uživateli, kteří jsou na tomto diskuzním fóru registrováni a přispívají svými dotazy a odpověďmi na jednotlivá témata, jež jsou umístěná v různých sekcích, které jsou zakládané uživateli.

2.1 Extrakce dat

Jako zdroj dat jsme použili matematické diskuzní fórum **http:// forum.matweb.cz** (viz výše). Získání samotných dat z diskuzního fóra bylo provedeno následujícím způsobem.

Nejprve bylo potřeba prostudovat html kód webových stránek. Zabýváme se zejména strukturou odkazů na jednotlivé části fóra (sekce, témata a příspěvky). Dále pak bloky html kódu, ze kterých je třeba získat pro nás důležitá data, např. id sekcí a loginy uživatelů. Podle takto nastudovaných html kódů webových stránek jsme vytvořili aplikaci v jazyce C#, která tento kód stránek prohledává. Mezi jednotlivými stránkami se aplikace pohybuje pomocí odkazů nalezených na základě regulárních výrazů. Jestliže stránka obsahuje data, která potřebujeme uložit, přistupujeme k nim opět pomocí regulárních výrazů.

Je třeba zdůraznit, že tvorba takových aplikací se liší pro každé diskuzní fórum, jelikož obvykle nemají stejný html kód a tedy ani strukturu. Například odkazy, s jejichž pomocí se aplikace na fóru pohybuje, mohou mít pro různá diskuzní fóra různou podobu.

2.1.1 Extrakce dat z matematického diskusního fóra

Vstupními daty pro aplikaci je url homepage, ve tvaru:

- **http://forum.matweb.cz/**

Tato url webové stránky se uloží do proměnné a provede se stažení a uložení obsahu stránek jako řetězce znaků:

```
if (url.ToLower().IndexOf("http:") > -1)
{
   System.Net.WebClient wc = new System.Net.WebClient();
  byte[] response = wc.DownloadData(url);
  sContents = System.Text.Encoding.UTF8.GetString(response);
}
```

Výpis 1: Stažení a uložení html stránky

Aplikace poté prochází html kód uložený v proměnné a na základě regulárního výrazu zjistí všechny odkazy ve tvaru:

- **Název sekce**

Struktura tohoto odkazu je pro všechny sekce stejná. Zdrojový kód pro nalezení příslušných odkazů odpovídající dané sekci:

```
// regularni vyraz, ktery najde odkazy vsech sekci na hlavni strance
string pattern = "(<a href=\"viewforum.php\\?id=)(\\d*)(\">)(.*)(</a>)";
//seznam nalezenych sekci, ulozenych v kolekci
MatchCollection mcSection = Regex.Matches(sContents, pattern, System.Text.RegularExpressions.
    RegexOptions.IgnoreCase);
//dokud pocet nalezenych retezcu je mensi jak i
while (mcSection.Count > i)
{
   //rozdeleni na jednotlive casti regularniho vyrazu (podle zavorek)
   GroupCollection gcSection = mcSection[i].Groups;
    ....
}
```

Výpis 2: Nalezení řetězce podle daného regulárního výrazu

Tyto odkazy tedy postupně procházíme. Před každým průchodem na stránku sekce se provede uložení dat do databáze. Tato stránka je opět prohledávána obdobným způsobem, opět se prochází html kód a opět pomocí regulárního výrazu se získávají bloky html kódu a odkazy, které značí témata v této sekci ve tvaru:

- **Název tématu**

Obdobným způsobem získáme id tématu a jeho název. K tomu je ovšem potřeba získat, kdy a kým bylo téma vytvořeno. Přejdeme na stránku s daným tématem. Zde se opět pomocí regulárních výrazů zjistí z určité části html kódu další potřebné informace, jako kdo a kdy téma založil. Dále zde zjistíme všechno, co se týká příspěvků napsaných v tomto tématu, jako je id příspěvku, kdo a kdy je napsal apod. Všechny tyto informace jsou opět uloženy do databáze. Jakmile jsou uloženy, pokračujeme v prohledávání dalšího tématu umístěného vdané sekci. Po ukončení prohledávání dané sekce se vrátíme k dalším sekcím umístěných na hlavní stránce a pokračujeme v jejich prohledávání. Vše se opakuje, dokud není zpracována poslední sekce. Jedná se tedy o několik vnořených cyklů.

Při prohledávání a zjišťování témat v sekci je nutno mít na paměti, že témata mohou být umístěna na více stránkách v dané sekci. Je tedy nutno projít všechny tyto stránky,

aby informace byly kompletní. Stejně tak i příspěvky v tématu mohou být rozděleny na více stránek.

Takovému zjišťování informací se odborně říká **Web Scraping**. Jedná se o získávání dat z html stránek pro jejich další využití. Nutno podotknout, že stahování dat proběhlo se svolením majitele diskuzního fóra. Staženo a uloženo do databáze bylo přibližně 250 000 řádků. Celé stahování trvalo přibližně dvě hodiny.

Algoritmus 1 Zjednodušeny algoritmus extrakce dat z diskuzního fóra

```
1:  procedure EXTRACTDATA(url homepage urlHomePage)
2:      for all urlSekce ∈ urlHomePage do
3:          Ulož potřebná data o sekci.
4:          for all urlTopic ∈ urlSekce do
5:              Ulož potřebná data o tématu.
6:              for all prispevek ∈ urlTopic do
7:                  Ulož potřebná data o příspěvku.
8:                  Ulož potřebná data o uživateli, který poslal daný příspěvek.
9:              end for
10:         end for
11:     end for
12: end procedure
```

2.2 Návrh databáze

Jako úložiště pro extrahovaná data bylo využito relační databázi MySql ve verzi 5.5.10. Na základě dat, které nás zajímají, jsme vytvořili ERD, podle něhož byla následně vytvořena databáze a v ní příslušné entity.

Při průchodu html stránkami mě zajímají tyto informace:

- **Sekce** - jednoznačné číslo sekce, název sekce
- **Téma** - jednoznačné číslo tématu, jednoznačné číslo sekce, do kterého téma spadá, název tématu, login zakladatele tématu a datum založení tématu
- **Příspěvek** - jednoznačné číslo příspěvku, jednoznačné číslo tématu, ve kterém byl příspěvek vložen, datum vložení příspěvku, login uživatele vkládajícího příspěvek
- **Uživatel** - jednoznačné číslo a login uživatele, datum registrace a jaký titul má uživatel na fóru

2.2.1 Lineární zápis entit a vztahů

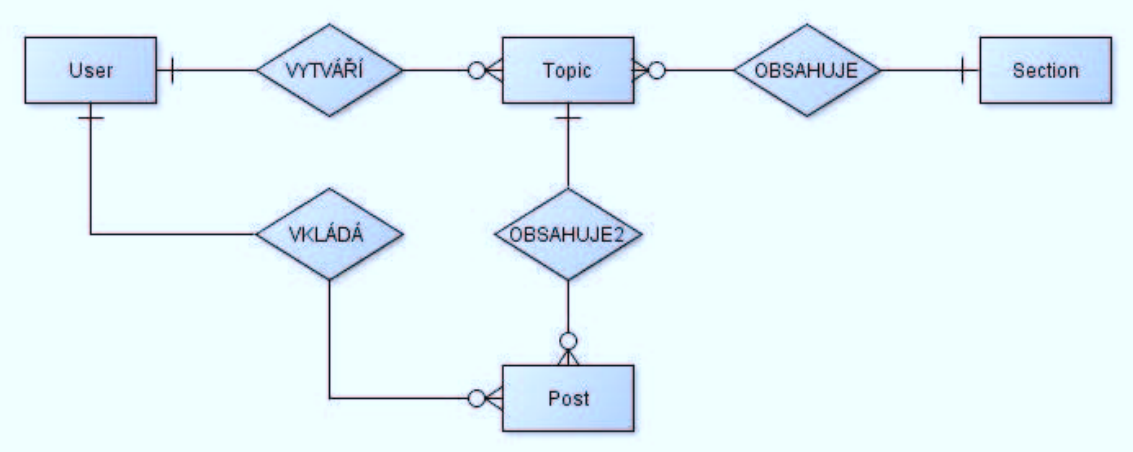
Popisuje entity a vztahy mezi entitami. Popis jednotlivých entit a jejich atributů je v předcházející kapitole (2.2).

Post (**postId**, *topicId*, *userId*, date)
Section(**sectionId**, name)
Topic (**topicId**, name, *sectionId*, start, *userId*)
User (**userId**, login, registration, title)

VYTVÁŘÍ (User, Topic)
VKLÁDÁ (User, Post)
OBSAHUJE (Section, Topic)
OBSAHUJE2 (Topic, Post)

2.2.2 ER diagram

Jedná se o často používaný model pro popis logické struktury dat na konceptuální úrovni. Popisuje objekty a jejich vztahy buď textovým zápisem nebo pomocí E-R diagramu.



Obrázek 1: ER diagram

2.2.3 Datový slovník

Soubor, který definuje strukturu a složení datové základny a obsahuje meta data potřebná pro správu dat. Datový slovník zahrnuje seznam všech datových objektu v databázi, jména a popis všech datových prvku a jejich vztahu, údaje o integritních omezeních atd.

Název	Typ	Velikost	Klíč	Null	Index	Popis
sectionId	int	10	PK	N	A	jednoznačná identifikace sekce
name	varchar	100	N	N	N	název sekce

Tabulka 1: Datový slovník - Section

Název	Typ	Velikost	Klíč	Null	Index	Popis
topicId	int	10	PK	N	A	jednoznačná identifikace tématu
name	varchar	300	N	N	N	název tématu
userId	int	10	FK	N	A	jednoznačné ID uživatele, cizí klíč z tab. User
sectionId	int	10	FK	N	A	jednoznačné ID sekce, cizí klíč z tab. Section
star	date		N	N	N	datum vložení tématu

Tabulka 2: Datový slovník - Topic

Název	Typ	Velikost	Klíč	Null	Index	Popis
postId	int	10	PK	N	A	jednoznačná identifikace příspěvku
topicId	int	10	FK	N	A	jednoznačné ID tématu, cizí klíč z tabulky Topic
userId	int	10	FK	N	A	jednoznačné ID uživatele, cizí klíč z tabulky User
date	date		N	N	N	datum vložení příspěvku

Tabulka 3: Datový slovník - Post

Název	Typ	Velikost	Klíč	Null	Index	Popis
userId	int	10	PK	N	A	jednoznačná identifikace uživatele
login	varchar	10	N	N	A	login uživatele
registration	date		FK	N	A	registrace uživatele
title	varchar	20	N	N	N	titul uživatele na fóru, moderátor apod.

Tabulka 4: Datový slovník - User

3 Zpracování extrahovaných dat do podoby vhodné pro soft shlukovací metody

V kapitole (2) je popsáno, jakým způsobem byla získána data pro testování, ovšem předtím než lze tato data použít, je třeba je upravit do podoby vhodné pro daný soft shlukovací algoritmus. Ze získaných dat jsme vytvořili vektorový model (3.1), který je např. základem metody fuzzy c-means. Jiné metody zase vyžadují, aby byla na vstupu matice podobnosti, získaná z vektorového modelu s využitím nějaké metody pro výpočet míry podobnosti (3.2) mezi objekty. To je případ metody GPSC nebo PoBOC.

3.1 Vektorový model

Vektorový model slouží k popsání objektu, dokumentu či dotazu prostřednictvím vektoru atributů (vlastností), kdy podobné dotazy mají podobný vektor [10]. Definice objektu vektorem:

- $o_i = (v_{i1}, v_{i2}, .., v_{in})$, kde v_{ij} představuje četnost atributu a_j.

Ze získaných dat jsme vytvořili **vektorové modely V(m,n)** reprezentované maticí **V(m $\times$ n)**, kde každý vektor (řádek této matice) představuje jednoho uživatele diskuzního fóra. Tento uživatel je popsán sadou atributů představující jeho vlastnosti. V rámci této diplomové bylo ze získaných dat vytvořeno prostřednictvím sql dotazů několik vektorových modelů pro další testování, a sice:

- Vektorový model s **průměrným časem odeslaných příspěvků.** U tohoto vektorového modelu je pouze jeden atribut, a to průměrný čas, ve kterém daný uživatel vkládá do diskuzí své příspěvky.
- Vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaných v daném časovém rozmezí** (např. první atribut, počet příspěvků uživatele od 0:00 do 0:59.
- Vektorový model, kde každý atribut představuje **počet příspěvku daného uživatele v dané diskuzi**. Vzhledem k tomu, že v databázi je uloženo přibližně 9000 uživatelů přispívajících v přibližně 33000 tématech, kde každý uživatel přispěl v jen jejich malé části, bude vektorový model obsahovat velký počet nulových hodnot. Jedná se o tzv. **řídká data**, která budou reprezentovaná v matici o velikosti 9000x32000 prvků. V takovém případě mluvíme o **řídké matici**.

3.1.1 Řídké matice

Jelikož by byl přístup k jednotlivým datům z důvodu velkého množství nulových prvků jak paměťově, tak i časově náročný, je třeba tuto matici vhodně upravit. Existuje celá řada metod pro reprezentaci řídkých matic jako např. ukládání po řádcích, plné uložení, pole spojových seznamů apod. [2]

V tomto případě jsme použili *metodu kompresního ukládání po řádcích* [2], která je reprezentována maticí *2 x počet všech „použitých" témat všech uživatelů*, tj. počet nenulových prvků ve vektorovém modelu, kde první řádek obsahuje sloupcový index původní řídké matice a druhý řádek hodnotu. Tato matice obsahuje již pouze nenulové prvky. Dále pak ještě potřebujeme pomocné pole, které obsahuje indexy z prvního řádku matice. Tyto indexy představují začátky řádků (tj. objektů) předchozí matice.

Př.

$$\begin{pmatrix} 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 2 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 2 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} =>$$

$=>$ $\{\{1,5,4,5,1,3,1,2,4,1\},\{1,2,1,2,1,1,1,1,2,1\}\}$ a $\{1,3,5,7,10\}$ = {{Sloupcové indexy obsahující nenulovou prvky},{hodnoty nenulových prvků}} a {indexy z pole sloupcových indexů reprezentující nové řádky}.

3.2 Typy podobnosti

Většina soft shlukovacích metod popsaných v této práci vyžaduje na vstupu data uložené v tzv. matici podobnosti S (similarity matrix). Tu lze získat kvantitativním pojetím podobnosti objektů. Pro dvojici objektů (o_i, o_j) je „s" stanovení vhodného předpisu přiřazující číslo $s(o_i, o_j)$ těmto objektům, splňující alespoň tyto požadavky [5].

- $s(o_i, o_j) \geq 0$
- $s(o_i, o_j) = s(o_j, o_i)$

Hodnota „s" vyjadřující míru podobnosti je tím větší, čím je podobnost daných objektů větší. Jestliže míra podobnosti může nabývat hodnot od 0 do 1, pak:

- pro $s(o_i, o_j) = 1$, jsou objekty stejné
- pro $s(o_i, o_j) = 0$, jsou objekty zcela odlišné

Podobnost objektů lze zjistit několika metodami rozdělenými do tří základních skupin [1]:

- Korelační míry
- Míry asociace
- Míry vzdálenosti
- Samotné míry podobnosti

3.2.1 Korelační míry

Jedná se o vzájemný lineární vztah mezi dvěma znaky či veličinami x a y. Jestliže se mění hodnota jedné veličiny, mění se i hodnota druhé veličiny. Míra korelace je vyjádřena koeficientem, který může nabývat hodnot od -1 do +1, kde:

- -1 značí antikorelaci, nepřímou závislost. Tedy čím více se zmenší hodnota první veličiny, tím více se zmenší hodnota veličiny druhé.
- +1 značí opačnou závislost. Tzn. jestliže se zvětší hodnota první veličiny, zvětší se hodnota i druhé veličiny.
- 0 značí nekorelovatelnost. Tedy neexistuje mezi veličinami žádný lineární vztah

K výpočtu korelace slouží Paersonův koeficient a Spearmanův korelační koeficient [1].

3.2.2 Míry asociace

Míry asociace bývají využívány pro binární proměnné, jež nabývají pouze hodnot 0,1. Při určování podobnosti prostřednictvím asociace dochází ke zjišťování, zda se pozorovaný znak v objektu vyskytuje v porovnání s objektem jiným.

Je dána kontingenční tabulka sledující vztahy mezi dvěma objekty o_1 a o_2: V tabulce

	$Objekt_i$		
		1	0
$Objekt_j$	1	a	b
	0	c	d

Tabulka 5: Tabulka popisující asociace mezi dvěma objekty [2]

jsou zahrnuty všechny možné kombinace znaků pro dva objekty [2].

Typy asociací:

- **Sochalův - Michenerův koeficient asociace**

$$s_{SM} = \frac{a+d}{a+b+c+d}$$

- **Russelův - Raovův koeficient asociace**

$$s_{RR} = \frac{d}{a+b+c+d}$$

- **Jaccardův koeficient asociace**

$$s_J = \frac{a}{a+b+c}$$

- **Rogersův a Tanimotův koeficient**

$$s_{RT} = \frac{a+d}{a+2b+2c+d}$$

Mezi další koeficienty asociace patří např. Sörensenův a Hammanův koeficient asociace [1].

3.2.3 Míry vzdálenosti

Nejběžnější a nejčastěji využívané pro vyjádření podobnosti mezi objekty jsou míry vzdálenosti. Vychází z geometrického modelu dat. Jestliže chceme porovnávat dva objekty vektorového modelu, pak souřadnice tvoří sada atributů tohoto objektu. Jednotlivé míry vzdálenosti mohou mít vliv na umístění objektů, což má za následek, že výsledná podoba shluků při použití různých metod se může lišit [5], [1].

Těmto mírám se také říká míry nepodobnosti. Abychom z nich dostali míry podobnosti, je třeba vzdálenost mezi dvěma objekty normovat, a to tak, že vzdálenost mezi dvěma objekty vydělíme největší vzdáleností mezi dvěma objekty z množiny dat. Podobnost pak získáme odečtením této normované vzdálenosti od jedné:

$$s(o_i, o_j) = 1 - d(o_i, o_j)/max_{\forall i,j} d(o_i, o_j)$$

- **Euklidovská vzdálenost**. Vzdálenost mezi dvěma body, dána Pythagorovou větou.

$$d(o_i, o_j) = \sqrt{(o_{i1} - o_{j1})^2 + ... + (o_{in} - o_{jn})^2}$$

- **Manhattanská metrika**. Metrika na množině R^n dána vztahem:

$$d(o_i, o_j) = |o_{i1} - o_{j1}| + ... + |o_{in} - o_{jn}|$$

- **Tětivová metrika**.

$$d(o_i, o_j) = \sqrt{2\Big(1 - \frac{\sum_{k=1}^{n} o_{ik} o_{jk}}{\sum_{k=1}^{n} o_{ik}^2 \sum_{k=1}^{n} o_{jk}^2}\Big)}$$

- **Minkovskiho metrika**. Pro tuto metriku platí, že pokud z = 1, jedná se o manhtattanskou metriku. Jestliže z = 2, mluvíme o euklidovské metrice. Čím je z větší, tím se více zdůrazňuje rozdíl mezi vzdálenějšími objekty.

$$d(o_i, o_j) = \sqrt[z]{\sum_{k=1}^{n} |o_{ik} - o_{jk}|^z}$$

Mezi další míry vzdálenosti patří Hammingova metrika, aj. [5], [1]

3.2.4 Samotné typy podobnosti

Jedná se o přímé metody výpočtu podobnosti, kdy hodnota může nabývat hodnotu od 0 do 1 (objekty jsou stejné).

- **Kosinova podobnost**. Jedná se o podobnost určenou pro nelineární poměr. Při použití kosinové podobnosti se získá kosinus úhlu, jenž svírá vektor mezi dvěma porovnávanými objekty [11].

$$s(o_i, o_j) = \frac{\sum_{k=1}^{n}(o_{ik}.o_{jk})}{\sqrt{\sum_{k=1}^{n}(o_{ik})^2.\sum_{k=1}^{n}(o_{jk})^2}}$$

- **Jaccardova podobnost**. Jedná se vlastně o poměr mezi průnikem a sjednocením atributů objektů. [20]

$$s(o_i, o_j) = \frac{|o_i \cap o_j|}{|o_i \cup o_j|}$$

Z podobnosti lze získat přímo normovanou vzdálenost ze vzorce:

$$d(o_i, o_j) = 1 - s(o_i, o_j)$$

4 Přehled soft shlukovacích metod

Clustering neboli shlukování je disciplína, jejíž kořeny sahají až do starověku. Shlukováním se zabýval již ve starověkém Řecku Aristoteles, když začal shlukovat zvířata do skupin na základě jejich podobnosti. Shlukování představuje velké množství metod, které umožňují analyzovat vícerozměrná data a sdružovat je do společných tříd. Jedná se tedy o shlukování objektů do skupin, neboli shluků podle různých charakteristik. Objekty v rámci jednoho shluků jsou pak odlišné od objektů, které náleží shlukům ostatním. Shlukování se vyskytuje v celé řadě disciplín jako např. biologie, medicina, chemie, ekonomie, informatika aj. V každé oblasti má svůj specifický název.

Shlukování můžeme rozdělit do dvou skupin podle toho, zda se pozorované objekty mohou objevit ve více shlucích či nikoli. Na tzv. **hard** shlukovací metody, u nichž každý objekt spadá pouze do jednoho shluku a **soft** shlukovací metody, častěji nazývané **překrývající** se shlukovací metody umožňující, aby daný objekt byl obsažen ve více shlucích současně. Může tedy docházet k překrývaní těchto shluků. Oproti hard shlukovacím metodám umožňují soft shlukovací metody pozorovat některé vlastnosti, např. v případě sociálních sítí, máme-li uživatelé jako objekty a skupiny jako shluky, lze jednotlivé objekty přiřadit k více shlukům, kdežto u klasických hard shlukovacích metod by objekt patřil pouze k jednomu shluku.

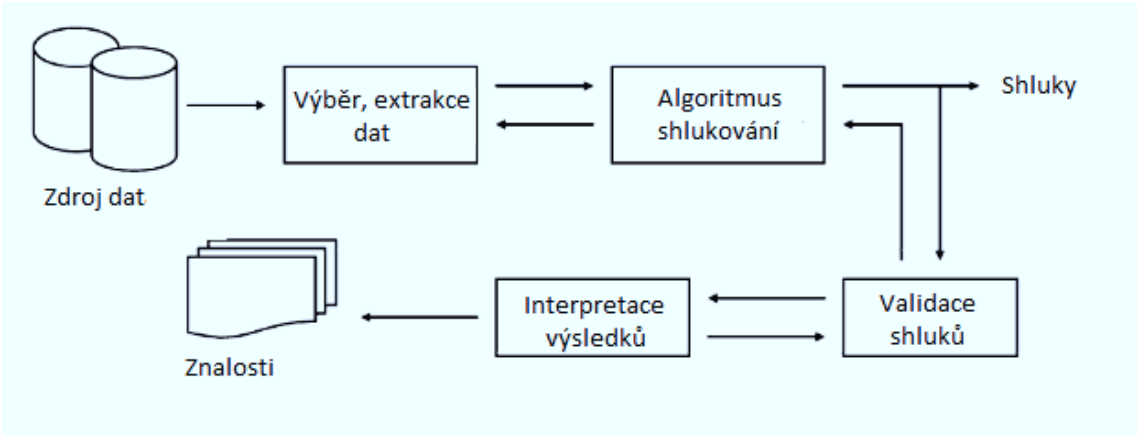
4.1 Soft shlukovací metody

Jak již bylo řečeno soft shlukovací metody umožňují, že jednotlivé objekty mohou patřit do více shluků, a tím může docházet k jejich překrývání. V dnešní době existuje celá řada soft shlukovacích metod. V následující části práce bude popsáno několik základních, z nichž některé budou následně implementovány.

Na níže uvedeném obrázku 2 lze pozorovat úlohy spjaté s životním cyklem shlukování. V první fázi je vybrán datový zdroj, z něhož jsou zpracována data. Této části odpovídá kapitola (2). Nad těmito daty je následně aplikovaný daný soft shlukovací algoritmus, díky němuž jsou vytvářeny shluky. Následuje fáze validace shluků, která má za cíl zjistit jejich kvalitu. Odpovídá-li kvalita shluků zadaným parametrům, následuje interpretace výsledků.

Rozdělení soft shlukovacích metod uvedených v této diplomové práci:

- Metody pracující s vektorovým modelem a s počáteční inicializací center shluků
  - Fuzzy c-means
  - Rough c-means
  - Rough fuzzy c-means
- Metody pracující nad grafem s matici podobnosti
  - Graph partitioning-based soft clustering
  - Pole based overlapping clustering



Obrázek 2: Proces shlukování

- Metody pracující s maticí podobnosti, s počáteční inicializací center shluků
  - Overlapping partitioning clustering

4.2 Metody pracují s vektorovým modelem

Jedná se o metody, které mají na vstupu algoritmu vektorový model, viz kapitola (3.1). Pro tyto metody je dále charakteristické, že před samotným vytvářením shluků se musí provést inicializace center shluků. Příslušnost objektů ke shlukům, popř. vzdálenost od center shluků se potom počítá s využitím metrik pro výpočet vzdálenosti, viz kapitola 3.2.3, kdy objekt patří do shluku, když je jeho příslušnost, popř. vzdálenost od centra shluku, větší než zadaná prahová hodnota ϵ.

Všechny tyto zde popsané algoritmy mají společnou ukončovací podmínku, a sice:

- Algoritmus dosáhne konečného počtu iterací, na které je přednastaven.
- Přestane docházet k výraznějším změnám, tzn. centra shluků se přestanou výrazněji měnit.

4.2.1 Inicializace počátečních center shluků

Pro inicializaci počátečních center shluků existuje celá řada metod. V další části je uvedeno několik základních, které jsou převzaté z článku [16].

- **Metoda náhodného výběru**. Jde o metodu založenou na náhodném výběru k center z m objektů. Nevýhodou těchto metod je, že nevedou ke zlepšení shlukování. Do této kategorie patří:
  - **R-SEL**. Klasická metoda, kde se každému centru náhodně vybere objekt z množiny m. Výběr je prováděn do doby, než je vybráno poslední centrum. V případě,

že je vygenerován objekt, který už byl dříve zvolen jako centrum, provede se generování aktuálního centra znovu.

– **R-MEAN**. Metoda náhodného výběru s využitím gaussova generátoru náhodných čísel, kde:

$$c_j = gaussianRandom(\overline{x}, \epsilon), kde : \quad (1)$$

$gaussianRandom(\overline{x}, \epsilon)$ = gaussianRandom(střední hodnota, odchylka) a $\overline{x} = \frac{1}{m}\sum_{i=1}^{m} x_i$ a ϵ je nějaká malá konstanta.

- **Metoda výběru prvních objektů**. Již podle názvu je patrné, že k shlukům se jako centra přiřadí prvních k objektů z množiny objektů O.

- **Metoda porovnávání s optimální množinou**. Jsou vybrány různé množiny počátečních center shluků. Ta množina, která se nejvíce podobá definované optimální množině, je nakonec použita.

- **Metody založené na optimalizaci vzdálenosti**. Jedná se o výpočet optimální vzdálenosti mezi centry shluku, což může mít za následek efektivnější výsledky shlukování.

  – **SCS metoda**. Prvnímu z k center je přiřazen první objekt z množiny objektů O. Postupně procházíme další objekty z množiny O, tj. $o_1, o_2, .., o_m$. Centrum dalšího shluku bude objekt o_i, jestliže pro všechny existující centra shluků platí, že $\|o_i - c_j\| > \epsilon$, kde c_j je aktuální centrum shluku a ϵ je prahová hodnota. Jestliže po projití všech objektů vstupní množiny O je počet center shluků menší než k, pak se snižuje prahová hodnota ϵ a začnou se znovu počítat zbývající centra shluku.

  – **KKZ metoda**. Prvnímu z k center je přiřazen objekt následujícím způsobem: $c_1 = max[\|o_i\|]$. Další centra shluků jsou inicializována tak, že pro každý objekt o_i je vypočítána vzdálenost k nejbližšímu počátečnímu centru $d_i = min[\|o_i - c_j\|]$ pro každé existující centrum. Objekt o_i, který má největší hodnotu d_i je další centrum shluku.

- **Metody založená odhadu hustoty**. Jedná se o metody založené na rozdělení vstupních dat prostřednictvím gaussova rozdělení. Jako centra vybíráme objekty z nejhustších oblastí vstupních dat. Prostřednictvím takto zvolených objektů jsou prostřednictvím dané metody vytvářeny kompaktní shluky.

  – **KR**. Jako centrum c_1 je zvolen nejvíce centrálně umístěný objekt. Každé následující centrum shluku c_j: pro každý nevybraný objekt o_i, spočítej vzdálenosti k dalším nevybraným objektům o_l, které mají blíže k o_i, než k jejich počátečním shlukům podle vzorce: $d_i = \sum_{l \neq i}\{max\{min[\|o_l - c_j\| : \text{for all } c_j] - \|o_l - o_i\|, 0\}\}$. Jestliže objekt který má největší hodnotu d_i je itý objekt, pak nastav $c_j = o_i$.

- **Další inicializační algoritmy**. Následují dva z mnoha nových inicializační algoritmů. Obě metody slouží k měření lokální hustoty bodu. Cílem je nalézt objekt

s nejvyšší lokální hustotou pro každý shluk jako centrum shluku. Jedná se o metody podobné těm, jež jsou založeny na odhadu hustoty [16]:

- K-nearest neighborhood
- ϵ-ball Measurment

4.2.2 K-means

Jedná se o jednoduchý algoritmus [17], vytvořený v roce 1967 Jamesem MacQueenem pro rozdělení m objektů do k shluků. K-means je hard shlukovací algoritmus, který každý objekt přiřadí právě do jednoho shluku. Z tohoto algoritmu vychází některé z následujících soft shlukovacích algoritmu, a sice Fuzzy c-means, Rough c-means a Rough fuzzy c-means.

Základem této metody je zvolení k center, buď náhodně nebo pomocí nějaké metody pro inicializaci center (4.2.1). Poté se prostřednictvím dané metody pro výpočet vzdálenosti (3.2.3) provede výpočet vzdálenosti jednotlivých objektů od zvolených center shluků. Objekt je následně přiřazen do shluku, jehož centrum je k objektu nejblíže.

V dalším kroku se provede výpočet nových center shluků jako těžiště objektů patřících k danému shluku, tj.:

$$c_j = \frac{1}{|C_j|} \sum_{o_i \in C_j} o_i \tag{2}$$

Poté se opět provede výpočet vzdáleností objektu od nových center a provede se jejich přiřazení ke shlukům. Tyto kroky se neustále opakují až do doby, než přestane docházet k podstatným změnám výsledku shlukování - jinými slovy, dokud se nepřestanou výrazně měnit centra shluků.

Cílem algoritmu je minimalizovat funkci:

$$J = \sum_{j=1}^{k} \sum_{i=1}^{m} \|o_i - c_j\|^2, kde: \tag{3}$$

$\|o_i - c_j\|^2$ představuje euklidovskou vzdálenost mezi objektem o_i a centrem c_j. Euklidovská vzdálenost může být nahrazena jinou metrikou pro výpočet vzdálenosti.

Algoritmus 2 K-means

1: **procedure** KMEANS(pocet_shluku k, množina_objektu O)
2: Vybranou metodou pro inicializaci center zvol centra shluků c z množiny objektů O.
3: **repeat**
4: Přiřaď jednotlivé objekty ke shlukům, jejichž centrum je nejblíže.
5: Vypočítej nové centra shluků podle jako těžiště objektů ve shluku (2)
6: **until** Centra shluků se nepřestanou výrazně měnit
7: **end procedure**

4.2.3 Fuzzy c-means

Metoda fuzzy c-means je základní a jednou z nejpoužívanějších soft shlukovacích metod, spadajících do kategorie fuzzy shlukovacích algoritmů. Byla vytvořena v roce 1973 J.C.Dunnem a v roce 1981 vylepšena Bezdekem. Metoda fuzzy c-means, stejně jako všechny soft shlukovací metody, umožňuje, aby každý objekt byl součástí více než jednoho shluku [6].

C-means umožňuje shlukování výhradně do kruhových tvarů (ve 2D). Základem této metody je určení příslušnosti každého objektu k danému shluku. Objekty, které leží blíže středu, mají větší stupeň příslušnosti než objekty, které leží na okraji shluků, do nichž se budou dané objekty shlukovat. Příslušnosti daných objektů jsou dány maticí příslušnosti. Je-li stupeň příslušnosti daného objektu ve shluku roven jedné, potom tento objekt patří výhradně do tohoto shluku. Před použitím fuzzy c-means je nutno zvolit počet shluků k, do kterých se objekty budou shlukovat [6].

Jako množinu vstupních dat máme zaanou matici $V(m,n) = (o_1, o_2, ..., o_m)$, tj. vektor objektů, kde $o_i \in R^n$ a $o_i = (v_{i1}, .., v_{in})$, tj. sada atributů daného vektoru. Matice V(m,n) je tedy vektor vektorů. Cílem fuzzy c-means algoritmu je minimalizovat fuzzy c-means funkci, která je formulována následujícím vztahem:

$$J(U,C) = \sum_{j=1}^{k}\sum_{i=1}^{m}(u_{ij})^p \|o_i - c_j\|^2, kde \tag{4}$$

- c_j představuje centrum shluku a C značí matici zvolených center shluků, kde $C = \{c_1, c_2, ..., c_k\}$
- $U = (u_{ij})_{m\times k}$ představuje matici příslušnosti, kde každý člen u_{ij} indikuje stupeň příslušnosti mezi datovým vektorem a shlukem c. Hodnoty matice by měly splňovat následující podmínky:
  - $u_{ij} \in [0,1], \forall i = 1, ..., m, \forall j = 1, ..., k$
  - $\sum_{j=1}^{k} u_{ij} = 1, \forall i = 1, ..., m$
- Exponent $p \in [1, \infty]$ určuje fuzziness koeficient shluků, jedná se o tzv. váhový exponent, který udává, v jaké míře se překrývají jednotlivé shluky.
- $\|o_i - c_j\|$ představuje vzdálenost objektu o_i od centra j-tého shluku C, kde:

$$c_j^{(b)} = \frac{\sum_{i=1}^{m}(u_{ij}^{(b)})^p o_i}{\sum_{i=1}^{m}(u_{ij}^{(b)})^p} \tag{5}$$

Algoritmus pracuje tak, že po zadání parametrů uživatelem, tj. do kolika shluků k se má shlukovat, váhový exponent p a prahová hodnota ϵ se provede jednou z metod pro výpočet počátečních center shluků , kapitola (4.2.1). Poté se provede výpočet matice příslušnosti:

$$u_{ij}^{(b+1)} = \frac{1}{\sum_{l=1}^{k} (\frac{d_{ij}}{d_{il}})^{\frac{2}{(p-1)}}} = \frac{1}{\sum_{l=1}^{k} (\frac{\|o_i - c_j\|}{\|o_i - c_l\|})^{\frac{2}{(p-1)}}}, \tag{6}$$

kde $d_{ij} = \|o_i - c_j\| > 0$ a jestliže pro $\forall i, j$ platí, že $d_{ij} = 0$, pak $u_{ij} = 1$ a $u_{lj} = 0, \forall l. \neq i$

Jakmile je spočítána matice příslušnosti, provede se přepočet nových center shluků podle vzorce (5), z nichž je vypočítána nová matice příslušnosti. Algoritmus pracuje do doby, než rozdíl těchto matic příslušnosti (aktuální a matice předcházející) spočítaný prostřednictvím metriky pro výpočet vzdálenosti není menší než zadaná prahová hodnota (centra shluků se již výrazně nemění), popř. dokud se neprovedl daný počet iterací.

Algoritmus 3 Fuzzy c-means

1: **procedure** FCM(pocet_shluku k, iteracni_krok b, vahovy_exponent p, prahova_hodnota ϵ, sada_objektu O)
2: Náhodně vyber centra shluků c z množiny objektů O.
3: **repeat**
4: Spočítej matici příslušnosti U podle vzorce (6) vzhledem k centrům shluku.
5: Proveď aktualizaci center shluků podle vzorce (5) s aktuální maticí příslušnosti U
6: Inkrementuj iterační krok
7: **until** ($\|U^{b+1}$ - $U^b\| \geq \epsilon$ nebo je dosaženo maximálního počtu iterací)
8: **end procedure**

4.2.4 Rough set-clustering

Rough sets clustering vychází z rough set teorie (dále jen RST), teorie hrubých množin, zveřejněné poprvé v 80. letech minulého století polským matematikem Zdislawem I. Pawlakem jako rozšíření stávající teorie množin sloužící pro analýzu nepřesných dat. Základem pro Pawlaka byla množina objektů v různých oblastech, nerozlišitelná podle dostupných metod pro jejich posuzování. Použití hrubých množin je vhodné a efektivní v případech, kdy chceme provádět analýzu nepřesných dat s nedostatečnými znalostmi, jež vedou k jejich nerozlišitelnosti [7],[13].

Rough set theory:
Základním pojmem RST [7],[13] je "Aproximační prostor" . Značí se "A" a je dán dvojicí A = (U,R), kde:

- U: neprázdná množina objektů, zvaná též univerzum

- R: relace nerozlišitelnosti, binární relace na U. Někdy bývá nazývána také jako relace ekvivalence. Jestliže $o_i, o_j \in U$ a $o_i R o_j$, kde o_i a o_j jsou nerozeznatelné v U.

Každá třída ekvivalence indukovaná relací nerozlišitelnosti R, tj. každá množina rozkladu R* = U/R je nazývána elementární množinou v aproximačním prostoru A. Aproximace prostoru může být zapsána též alternativním způsobem, a sice A = (U, R*). Každé sjednocení elementárních množin se nazývá rozlišitelná množina. Opakem je potom nerozlišitelná množina, představující hrubou množinu v aproximačním prostoru. Elementární množinu pro každé o patřící do množiny U značíme $[o]_R$. Jestliže $O \subseteq U$, pak jsou definovány dvě množiny, a sice dolní a horní aproximace množiny O:

- Dolní aproximace: $\underline{R}O = \{o \in U | [o]_R \subseteq O\}$
- Horní aproximace: $\overline{R}O = \{o \in U | [o]_R \cap O \neq 0\}$

Množina $B = \overline{R}O - \underline{R}O$ se nazývá hranice množiny. Dále pak definujeme pozitivní obor a negativní obor množiny O, kde:

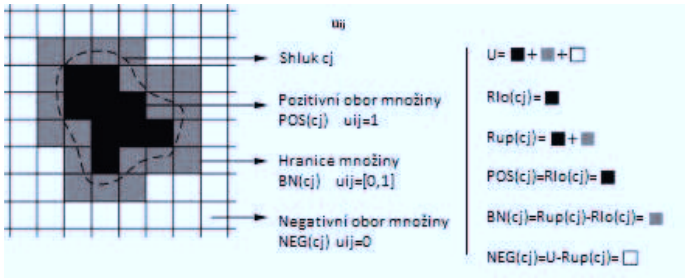
- Pozitivní obor množiny O$POS_R = \underline{R}O$
- Negativní obor množiny O$NEG_R = U - \overline{R}O$

Pozitivní obor množiny O obsahuje všechny objekty množiny O, kdežto negativní obor množiny O obsahuje objekty, které nepatří do množiny O. Dohromady dávají množinu všech prvků Univerza.

Přesnost a hrubost množiny O se pak značí $\alpha_R(O)$, respektive $\mu_R(O)$, kde:

- $\alpha_R(O) = \frac{card(\underline{R}O)}{card(\overline{R}O)}$
- $\mu_R(O) = 1 - \alpha_R(O)$

Pro přesnost a hrubost množiny O platí, že $\alpha_R(O) \geq 0$ a $\mu_R(O) \leq 1$



Obrázek 3: Aproximace hrubých množin[13].

Více [7],[13].

Rough cluster, nebo-li hrubý shluk, je definován podobným způsobem, pomocí spodní a horní aproximace. Spodní aproximace hrubého shluku obsahuje objekty, které jsou součástí pouze tohoto jednoho shluku. Horní aproximace hrubého shluku obsahuje i ty objekty, které jsou součástí i jiných shluků.

4.2.5 Rough c-means

Jedná se o modifikovaný c-means algoritmus [12] využívající koncepci rough set množin a dolní a horní aproximace hrubých množin. Každý shluk je zastoupený dolní a horní aproximací. Každý objekt může patřit buď k dolní aproximaci jednoho shluku, tzn. patří pouze k tomuto shluku, nebo může patřit k více horním aproximacím různých shluků, tzn. objekt patří do více shluků.

Na vstup je tedy přivedena stejně jako u fuzzy c-means množina objektů a prahová hodnota. Kromě těchto parametrů je však nutné zadat w_{low}, kde $w_{high} = 1 - w_{low}$, představující relativní význam dolní a horní meze.

Stejně jako u fuzzy c-means je na začátku provedena inicializace center k shluků. Místo matice příslušnosti je zde ovšem počítána matice vzdálenosti mezi centry shluků z množiny C a objekty z množiny O, tj. $\|o_i - c_j\|$.

Následuje přiřazení objektu k dolní a horní aproximaci. Objekt, který patří do horní aproximace nějakého shluku, nemůže patřit do žádné dolní aproximace. Přiřazení do horní či dolní aproximace se provádí tak, že se vybere vzdálenost objektu o_i k nejbližšímu centru c_j. Poté se prochází vzdálenosti mezi tímto objektem o_i a ostatními centry c_l. Jestliže $\|d_{il} - d_{ij}\|$ bude menší jak prahová hodnota, objekt bude přiřazen do horní aproximace shluku s centrem c_j i s centrem c_l. V opačném případě bude přiřazen do dolní aproximace s centrem c_j. Poté se provede přepočet nových center shluku podle vzorce (7).

Algoritmus končí obdobně jako v předchozím případě, a sice pokud se provedl maximální počet iterací nebo pokud přestane docházet k podstatným změnám výsledku shlukování, tedy dokud se nepřestanou výrazně měnit centra shluků.

$$c_j = \left\{ \begin{array}{ll} \omega_{low} \frac{\sum_{o_i \in \underline{R}(O_j)} o_i}{|\underline{R}(O_j)|} + \omega_{up} \frac{\sum_{o_i \in B(O_j)} o_i}{|B(O_j)|}, & \text{if } \underline{R}(O_j) \neq \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in B(O_j)} o_i}{|B(O_j)|}, & \text{if } \underline{R}(O_j) = \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in \underline{R}(O_j)} o_i}{|\underline{R}(O_j)|}, & \text{ostatní} \end{array} \right\} \tag{7}$$

4.2.6 Rough fuzzy c-means

Jedná se o fuzzy verzi předcházejícího algoritmu [12]. Oproti předcházejícímu rough c-means algoritmu se liší v tom, že nepracuje s maticí vzdálenosti objektů od center, ale s maticí příslušnosti vycházející z fuzzy c-means, umožňující více robustnější shlukování.

I zde se provádí rozdělení objektů do horní a dolní aproximace shluků, jen s tím rozdílem, že zde se nejdříve porovnají příslušnosti objektu ke shlukům u_{ij}, z nichž je vybrána ta největší. Od té jsou následně odečítány zbývající příslušnosti tohoto objektu k ostatním shlukům, jestliže bude výsledná hodnota menší než prahová hodnota ϵ, pak

Algoritmus 4 Rough c-means

procedure FCM(pocet_shluku k, iteracni_krok b, vahovy_exponent p, prahova_hodnota ϵ, sada_objektu O)
  Proveď inicalizaci center shluků c z množiny objektů O.
  repeat
    for objekt ($o_i \in$ O) **do**
      Spočítej vzdálenosti objektů ke všem centrům jako matici vzdáleností D = $\{d_{ij}\}$, $j \in \{1, .., k\}$
      $d_{ij} <=$ minimální vzdálenost od centra pro daný objekt
      for každý shluk c_l, kde j $\neq$ l **do**
        if d_{il} - $d_{ij} < \epsilon$ **then**
          Přiřaď objekt o_i k oboum horním aproximacím $o_i \in \overline{R}O_j$, $o_i \in \overline{R}O_l$ a o_i nemůže být členem žádné dolní aproximace
        else
          Přiřaď objekt o_i k dolní aproximaci $o_i \in \underline{R}O_j$
        end if
      end for
    end for
    Spočti nová centra shluků podle (7)
  until (Dochází k výrazným změnám nebo je dosaženo maximálního počtu iterací)
end procedure

objekt bude spadat do dolních aproximací daných shluků, tj. pokud $u_{ij} - u_{ik} < \epsilon$, pak $o_i \in R\overline{R}(O_j)$ a $o_i \in \overline{R}(O_k)$.

Aby objekt o_i patřil do dolní aproximace shluku, nesmí patřit do žádné horní aproximace jakéhokoliv shluku.

Algoritmus je tedy stejný jako předcházející, jen je zde rozšířený o práci s fuzziness koeficientem nutným pro výpočet matice příslušnosti U, viz vzorec (6) a pro výpočet nových center shluků (26).

$$c_j = \left\{ \begin{array}{ll} \omega_{low} \frac{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p o_i}{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p} + \omega_{up} \frac{\sum_{o_i \in B(O_j)} u_{ij}^p o_i}{\sum_{o_i \in B(O_j)} u_{ij}^p}, & \text{if } \underline{R}(O_j) \neq \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in B(O_j)} u_{ij}^p o_i}{\sum_{o_i \in B(O_j)} u_{ij}^p}, & \text{if } \underline{R}(O_j) = \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p o_i}{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p}, & \text{ostatní} \end{array} \right\} \quad (8)$$

4.3 Metody pracující nad grafem, využívající matici podobnosti

Všechny následující algoritmy pracují s maticí podobnosti získanou z množiny objektů O prostřednictvím některé z metod pro výpočet podobnosti mezi objekty (3.2). Dalším společným rysem je, že z matice podobnosti je vytvářen graf G, s nímž se dále pracuje.

Algoritmus 5 Rough fuzzy c-means

procedure FCM(pocet_shluku k, iteracni_krok b, vahovy_exponent p, prahova_hodnota ϵ, sada_objektu O)
  Proveď inicalizaci center shluků c z množiny objektů O.
  repeat
    for objekt ($o_i \in$ O) **do**
      Spočítej příslušnosti objektů ke všem centrům jako matici příslušnosti U = $\{u_{ij}\}$, $j \in \{1, .., k\}$
      u_{ij} <= maximální hodnota příslušnosti pro daný objekt
      for každý shluk c_l, kde j $\neq$ l **do**
        if u_{ij} - $u_{il} < \epsilon$ **then**
          Přiřaď objekt o_i k oběma horním aproximacím $o_i \in \overline{R}O_j$, $o_i \in \overline{R}O_l$ a o_i nemůže být členem žádné dolní aproximace
        else
          Přiřaď objekt o_i k dolní aproximaci $o_i \in \underline{R}O_j$
        end if
      end for
    end for
    Spočítej nová centra shluků podle (26)
  until (Dochází k výrazným změnám nebo je dosaženo maximálního počtu iterací)
end procedure

4.3.1 Graph Partitioning-based Soft Clustering Algorithm

Popis tohoto algoritmu byl převzatý z článku [8]. GPSC je efektivní soft shlukovací algoritmus založený na grafovém modelu. V tomto algoritmu je nejdříve z množiny vstupních dat vytvořen graf, a pak na základě metody pro rozdělování dochází k rozdělení objektů, respektive vrcholů do shluků. Poté je definován stupeň příslušnosti jednotlivých vrcholů grafu k centru shluků a vztah k sousedním shlukům.

Základem algoritmu je vytvoření **matice podobnosti** S s využitím metody definující podobnost objektů, viz kapitola (3.2).

Pro libovolné $\epsilon \in [0, 1]$, množina $S_\epsilon\{(o_i, o_j)|s(o_i, o_j) > \epsilon\}$ se nazývá ϵ - množina matice S, kde ϵ je prahová hodnota. Pomocí prahové hodnoty mohou být některé podobnosti mezi objekty eliminovány. Pokud jejich podobnost bude menší než zadaná prahová hodnota ϵ, pak lze identifikovat odlehlé hodnoty.

Prahová hodnota ϵ by měla být zvolena s ohledem na testovací množinu dat. Před spuštěním samotného algoritmu by mělo dojít k testování zkoumaných dat, po kterém by mělo dojít k volbě prahové hodnoty ϵ. Ta by neměla být příliš vysoká ani nízká, aby nedošlo k tomu, že data budou brána jako odlehlá a jejich počet bude vysoký. Ovšem testování rozsáhlých testovacích množin může být časově náročné, pak se volí ϵ náhodně.

Poté je z matice podobnosti $S_{m\times m}$, vytvořen graf G(V,E), kde V je množina vrcholů odpovídající množině dat O a množina E představuje množinu hran odpovídající vztahům mezi objekty. Jestliže platí, že $s(o_i, o_j) > \epsilon$, pak hrana $e(o_i, o_j) \in E$. Jestliže neexistuje

hrana spojující vrchol (objekt) s dalšími vrcholy (objekty), jedná se o odlehlou hodnotu. Takov graf je nazýván **base-graph**.

Graph partitioning metoda prostřednictvím grafového modelu popisuje *shlukovací problém*, při kterém můžeme dostat počáteční výsledek shlukování rozdělující m vrcholů do k shluků, kde:

- **Vstup:** Graf G(V,E) s váhami hran a vrcholů a parametr k představující počet shluků
- **Výstup:** Rozdělení vrcholů do k shluků takovým způsobem, aby součet vrcholů v každé množině byl stejný a součet vah hran mezi množinami je minimalizovaný.

Množina P grafu G obsahuje všechny „cut“ hrany představující hranu mezi dvěma shluky.

Definice 4.1 *Jestliže množina $V'_k \subset V_k$ a $\forall e = (v_i, v_j)$, kde $v_i \in V'_k, v_j \in V_k$, pak V'_k je množina hlavních vrcholů shluku C_k. Vrcholy $(v_i, v_j) \in V'_k$ jsou hlavní vrcholy.*

Definice 4.2 *Jestliže $e = (v_i, v_j) \in P, v_i \in V_k, v_j \notin V_k$, pak v_i jsou okrajové vrcholy C_k. Množina V''_k je množina všech okrajových vrcholů shluku C_k, pro níž platí, že $V''_k \subset V_k$. Tyto vrcholy $(v_i, v_j) \in V''_k$ jsou vrcholy okrajovými.*

Jestliže podmnožina vrcholů V'_k je součástí V_k, pak všechny vrcholy V'_k mají hrany pouze s vrcholy ve V_k. Tyto vrcholy jsou **hlavními vrcholy** V_k.

Vrcholy mající hrany s vrcholy z množinyV_k, které nepatří do množiny V_k a s vrcholy ze sousedních shluků, se nazývají **okrajové vrcholy**, vrcholy spadající do V''_k. Pomocí těchto dvou definic 4.1 a 4.2 můžeme snadno najít ty objekty, které mají odlišně shluky (hlavní vrcholy) a ty objekty, které jsou pro různé shluky splývající (okrajové vrcholy).

Platí, že $V_k = V'_k + V''_k$

Pro každý vrchol, jehož příslušnost se rovná jedné platí, že patří pouze do jednoho shluku. Příslušnost vrcholu ke shlukům se zjistí prostřednictvím hran s vahou, které tyto vrcholy spojují. Jestliže je vrchol spojen s dalšími vrcholy patřící do různých shluků, tedy je spojen s ostatními shluky, jedná se o okrajový vrchol, jehož příslušnost lze spočítat jako poměr váhy vrcholu vedoucí z vrcholu do daného vrcholu v_i se součtem všech hran vedoucí z vrcholu v_i.

Je-li dána množina hran E_i spojená s vrcholem v_i, pak váha pro každou hranu se značí s_{ij}, jedná se vlastně o podobnost mezi dvěma objekty.

Následující funkce popisuje příslušnost vrcholu ke shluku jako poměr součtu hran vedoucích z daného vrcholu do vrcholu v_i a součtu všech hran vrcholu v_i :

$$m\left(C_k, v_i\right) = \frac{\sum_{o_j \in V_k} s_{ij}}{\sum_{\forall o_j} s_{ij}} \tag{9}$$

Pozn. mluvíme-li o vrcholech v, máme tím na mysli vlastně objekty o a naopak, tj. $o_i = v_i$

Vztah mezi shluky lze měřit pomocí vah „cut“ hran a všemi váhami hran.

Funkce popisující vztah mezi dvěma shluky, který je dán jako poměr hran spojující dva shluky a všech hran:

$$m\left(C_i, C_j\right) = \frac{\sum_{i \in V_i, j \in V_j} s_{ij}}{\sum_S s_{ij}} \tag{10}$$

Použitím předešlých dvou vztahů (9) a (10) získáme dvě matice, a sice matici příslušnosti $m \times k$ a matici vztahů $k \times k$, kde m je počet objektů a k počet shluků.

Algoritmus 6 GPSC algoritmus

1: **procedure** FCM(matice_podobnosti S, prahova_hodnota ϵ)
2: Z matice podobnosti S vytvoř graf G(V,E), kde hrana(u,v) patří do množiny E, je-li váha hrany větší než prahová hodnota α.
3: Pomocí nějaké metody rozdělení grafu proveď rozdělení m vrcholů do k shluků.
4: Vypočítej stupeň příslušnosti každého vrcholu ke shluku a ulož do matice příslušnosti.
5: Z matice příslušnosti a vztahů nalezni skryté shluky.
6: **end procedure**

4.3.2 Poles Based Overlappnig Clustering

Popis algoritmu je převzatý z [4]. Jedná se o překrývající algoritmus, kde počet shluků není dopředu znám. Na základě matice podobnosti, vytvořené z množiny objektů, jsou vytvářeny malé homogenní množiny objektů nazývané póly, ke kterým jsou následně přiřazovány tyto objekty. Poskytuje kompromis mezi hard shlukováním a fuzzy shlukováním, poskytující vhodný počet shluků, které se překrývají.

Definice 4.3 *Jestliže je dána množina objektů $O = \{o_1, o_2, .., o_n\}$, matice podobnosti S je dána jako $m \times m$, pak **graf podobnosti** $G_s(V, E)$ je množina vrcholů V (objektů matice podobnosti) a množina hran E takových, že $(v_i, v_j) \in E$ (v_i je spojená v_j) právě, když:*

$$s(o_i, o_j) \geq max\{\frac{1}{n} \sum_{o_k \in O} s(o_i, o_k), \frac{1}{n} \sum_{o_k \in O} s(o_j, o_k)\} \tag{11}$$

Hrana mezi vrcholy v_i a v_j(objekty o_i a o_j) existuje, jestliže jejich podobnost je větší než průměrná podobnost mezi v_i a celou množinou objektů a zároveň průměrná podobnost mezi v_j a celou množinou. Kvůli tomu nedochází ke specifikaci prahových hodnot odpovídajících minimální hodnotě podobnosti.

Definice 4.4 *Pól P_k je podmnožina množiny objektů O takových, že podgraf $G_s(P_k, E(P_k))$ je klika, tzn. že $\forall v_i \in P_k, \forall v_j \in P_k$ a $(v_i, v_j) \in E(P_k)$, kde $E(P_k)$ je množina vrcholů (v_i, v_j) takových, že $v_i \in P_k$ a $v_j \in P_k$*

Konstrukce pólů vyžaduje vytvoření množiny klik v grafu podobnosti. To je v tomto případě řešeno heuristicky. Při vytváření kliky se začíná od jednoho vrcholu. K tomuto vrcholu se opakovaně přidávají nejbližší sousední vrcholy, dokud není nalezen vrchol spojený s každým vrcholem. Množina pólu je pak dána opakováním tohoto postupu.

První vybraný vrchol v^1 má nižší průměrnou podobnost s ostatními objekty. Je spojený alespoň s jedním vrcholem z množiny objektů.

$$v^1 = Argmin_{v_i \in D} \frac{1}{|V|} \sum_{v_j \in V} s(v_i, v_j), \tag{12}$$

kde D je množina vrcholů mající alespoň jeden připojený vrchol.

Další vrcholy $\{v^2, .., v^l\}$ jsou vybírány tak, aby se snížila podobnost s již vytvořenými póly.

$$v^k = Argmin_{v_i \in D} \frac{1}{v-1} \sum_{l=1,...,k-1} \frac{1}{|P_l|} \sum_{v_j \in P_l} s(v_i, v_j) \tag{13}$$

Proces se zastaví, když součet této rovnice je větší než průměrná podobnost k celé množině objektů. Tímto způsobem je určen **počet pólů a tedy i počet shluků**.

Definice 4.5 *Je dána množina $O = \{o_1, .., o_i\}$, množina pólů $P = \{p_1, .., p_l\}$ a matice příslušnosti U, která je dána jako $m \times l$ (velikost množiny klik). Přiřazení objektů k pólům je důležité při tvorbě shluků. Pro daný objekt o_j, $P_{j,1}$ je pak nejvíce podobný pól pro $o_j (P_{j,1} = Argmax_{P_i \in P} u(P_i, o_j))$, $P_{j,2}$ je druhý nejvíce podobný pro o_j atd.*

Metoda Assign $(o_j, P_{j,k})$ je použita k testování právě, když je objekt o_j je přiřazen k pólu $P_{j,k}$, právě, když je splněna jedna z následujících vlastností.

- *k = l, tzn. že lze přiřadit každému objektu alespoň jeden pól.*
- *$1 < k < l, u(P_{j,k}, o_j) \geq \frac{u(P_{j,k-1}, o_j) + u(P_{j,k+1}, o_j)}{2} a \forall k' < k, assign(o_j, P_{j,k'})$, tzn. přiřadit objekt o_j k pólu $P_{j,k}$ na základě zvážení podobnosti s předchozím pólem $P_{j,k-1}$ a dalším pólem $P_{j,k+1}$*

Posledním krokem je **hierarchické řazení skupin** umožňující kontrolu konečného počtu shluků složených z množiny objektů skupin. K hierarchickému uspořádání je využito hierarchické aglomerativní metody *single-link*. Je dána množina dříve vytvořených skupin $C = \{C_1, C_2, .., C_l\}$, kde $C_i = \{o_j$ přiřazených k $P_i\}$. Jelikož je matice podobnosti normalizována, $\forall o_i \in O, s(o_i, o_i) = 1$ je dána podobnost mezi dvěma skupinami:

$$sim(C_k, C_m) = \frac{1}{|C_k|.|C_m|} \sum_{o_i \in C_k} \sum_{o_j \in C_m} s(o_i, o_j) \tag{14}$$

Dvě nejpodobnější skupiny jsou aglomerovány. Tento proces je opakován dokud nedostaneme pouze jeden shluk. Organizace využívá binárního stromu, jehož listy odpovídají počáteční množině skupin.

Algoritmus 7 PoBOC algoritmus

1: **procedure** POBOC(matice_podobnosti S, množina objektů O)
2: Z matice podobnosti S vytvoř graf podobnosti G_s(V,E).
3: Vytvoř **množiny pólů P** = $\{p_1, p_2, .., p_n\}$, kde $\forall i \in \{1, 2, ..., l\}$ a $P_i \subseteq V$ podle 4.5.1
4: Vytvoř **matici příslušnosti U**, kde u$(P_i, o_j) = \frac{1}{|P_i|} \sum_{o_k \in P_i} s(o_i, o_j)$, kde součet hodnot příslušnosti daného objektu ve všech pólech nesmí být roven 1.
5: Z matice příslušnosti a vztahů nalezni skryté shluky.
6: **for all** $o_j \in$ V **do**
7: volej metodu assign(o_j,P)
8: **end for**
9: Necht'C je množina skupin(shluků) $\{C_1, C_2, .., C_n\}$ takových, že $C_i = \{o_j \in O | o_j$ je přiřazeno k P_i. Vytváření hierarchického uspořádání C.
10: **end procedure**

pozn. Někde je podobnost uváděna jako podobnost dvou objektů, tj. s(o_i, o_j) a někde jako podobnost dvou vrcholů, tj. s(v_i, v_j) $\Rightarrow$ mluvíme-li o vrcholu, myslíme tím vlastně objekt a naopak.

4.4 Metody pracující s podobností a s počáteční inicializací center shluků

V této části je uveden algoritmus, který je kombinací předcházejících dvou typů metod. Algoritmus zde pracuje s maticí podobnosti, která je na rozdíl od předešlých algoritmů, které využívají matici podobnosti, získávána výhradně ze vzdálenosti objektů a převedena na podobnost vztahem 15. Dále využívá počáteční inicializace center shluků náhodným výběrem. Ty jsou pak dále přepočítávány a měněny do doby než je nalezeno optimální shlukování.

4.4.1 Overlapping partitioning cluster algoritmus (Non-exhaustive clustering)

Jedná se o heuristický překrývající se shlukovací algoritmus pracující do doby než nalezne uspokojující výsledky (převzatý z [9]), který umožňuje objekt zařadit jak do více shluků, tak do žádného shluku. Dané objekty tedy nemusí náležet žádnému shluku. Další vlastností tohoto algoritmu je maximalizace průměrného počtu objektů ve shluku a maximalizace vzdálenosti center mezi jednotlivými shluky. Součástí tohoto algoritmu je tedy hledání center shluků mezi objekty, jenž jsou od sebe nejvíce vzdáleny.

Podobně jako u většiny výše uvedených algoritmů je i zde nutné zadat počet shluků, do kterých se budou objekty shlukovat. Navíc je zadána prahová hodnota, určující hranice shluku. Díky tomu může právě objekt spadat do jednoho či více objektů nebo pokud se objekt nachází ve velké vzdálenosti, nemusí spadat do žádného shluku. Jedná se tedy o tzv. Non-exhaustive clustering (neúplné shlukování).

Prvním krokem je **výpočet matice vzdáleností a podobnosti mezi objekty**. Nejdříve se provede výpočet vzdáleností mezi jednotlivými objekty, např. euklidovskou, manhattonovskou či jinou metodou, z níž se následně provede výpočet podobnosti mezi těmito objekty. Tyto podobnosti jsou poté normalizovány. Mají tak hodnotu podobnosti 0 v případě, že jsou od sebe hodně vzdáleny a 1, pokud jsou u sebe blízko, tj. atributy těchto objektů jsou totožné. Výpočet normalizované podobnosti vychází ze vzorce:

$$s_{ij} = 1 - min\{d_{ij}, d_{if}\}/d_{if}, kde: \tag{15}$$

- s_{ij} je podobnost mezi objekty o_i a o_j
- d_{ij} je vzdálenost mezi objekty o_i a o_j
- d_{if} je top 5% percentil vzdáleností všech objektů

Z toho vyplývá, že je-li vzdálenost d_{ij} mezi objekty i a j větší než d_{if}, pak podobnost $s_{ij} = 0$.

Prostřednictvím **prahové hodnoty** ϵ určujeme, zda daný objekt o_i patří do shluku C_j s centrem shluku c_j, tj. pokud je splněno, že $s_{ij} > \epsilon$, kde:

$$s_{ij} = 1 - min\{d_{ij}, d_{if}\}/d_{if}, \text{ tj. } d_{if} - min\{d_{ij}, d_{if}\} > \epsilon \times d_{if}, kde: \tag{16}$$

Matice podobnosti je zde získávána z matice vzdálenosti také z toho důvodu, že matice podobnosti je zde rozšířena o další tři sloupce, k jejichž zisku se využívá právě vzdálenosti mezi objekty a centry shluků, a sice:

- **Crowding value**

$$Cv(o_i) = no_i/maxv, kde: \tag{17}$$

  - no_i představuje počet objektů ve shluku se středem o_i a maxv je největší ze všech no_i.
  - Čím je $Cv(o_i)$ větší, tím více objektů bude ve shluku, pokud bude o_i vybrán jako centrum shluku. Je-li zvolen jako centrum shluku objekt s větším Cv, pak počet objektů ve shluku je zvýšen. $Cv(o_i)$ tedy slouží jako odhad počtu objektů patřících ke shluku s centrem shluku ci.

- **Mdv**

$$Mdv(o_i) = ndo_i/maxd \tag{18}$$

  - kde ndo_i je vzdálenost objektu o_i k nejbližšímu centrálnímu objektu (centru shluku), maxd je maximální vzdálenost všech ndo_i.

- **Center recommend fucntion**

$$CRF(o_i) = w_1 \times Cv(o_i) + w_2 \times Mdv(o_i) \tag{19}$$

- Funkce CRF na základě $Cv(o_i)$ a $Mdv(o_i)$ určuje, které objekty, jenž nejsou centry shluků, mohou být použity právě jako centra.
- V závislosti na potřebách lze nastavit váhy w_1 a w_2.

Výhodnější je vybrat objekt s vyšší CRF hodnotou, než s nižší. Proto je pravděpodobnost výběru objektu, který není centrem jako centrum shluku přímo úměrné CRF. **Pravděpodobnost výběru:**

$$Prob(o_i) = \frac{CRF(o_i)}{\sum_{i=1}^{n} CRF(o_i)} \tag{20}$$

Jakmile je vytvořena matice příslušnosti, provede se výběr počátečních k center. Poté se provede přiřazení objektů do shluků s využitím prahové hodnoty a provede se update Mdv, Cv a CRF pro všechny objekty o_i.

Dalším krokem je výpočet **cílové hodnoty aktuálního shlukování**.

$$Obj_{current} = w_1 \times min[Mdv(o_{c1}), ..., Mdv(o_{ck})] + w_2 \times \sum_{i=1}^{k} \frac{Cv(o_i)}{k}, kde: \tag{21}$$

- $Cv(o_i)$ určuje kolik objektů bude patřit k danému shluku
- Vzdálenost mezi centry shluků je definována jako $Min_{i \neq j} d_{ci,cj}$
- $Mdv(o_{ci})$ označuje normalizovanou vzdálenosto_{ci} na nejbližší objekt, který je centrem shluku. Vzdálenost mezi shluky může být přepsána $Min[Mdv(o_{c1}), ..., Mdv(o_{ck})]$.

Poté se iterativně upravují aktuální, které mají za cíl vylepšit výsledky shlukování. Objekt na základě CRF, který není centrem, nahradí dočasně objekt, jenž je centrem shluku. Poté je vypočítána cílová hodnota nového shluku. Je-li nová cílová hodnota větší než stará, nové centrum nahradí staré centrum shluku trvale. Tento proces se opakuje, dokud cílové hodnoty nekonvergují nebo počet iterací dosáhl předem určeného počtu.

Případy nahrazení center shluků novým objektem:

První případ

- **Situace:** o_i je objekt, o_s je nové centrum, které nahradí staré centrum o_j. $ndo_i = d_{ij}$ a $d_{ij} > d_{is}$.
- **Výsledek:** $ndo_i = d_{is}$.

Druhý případ

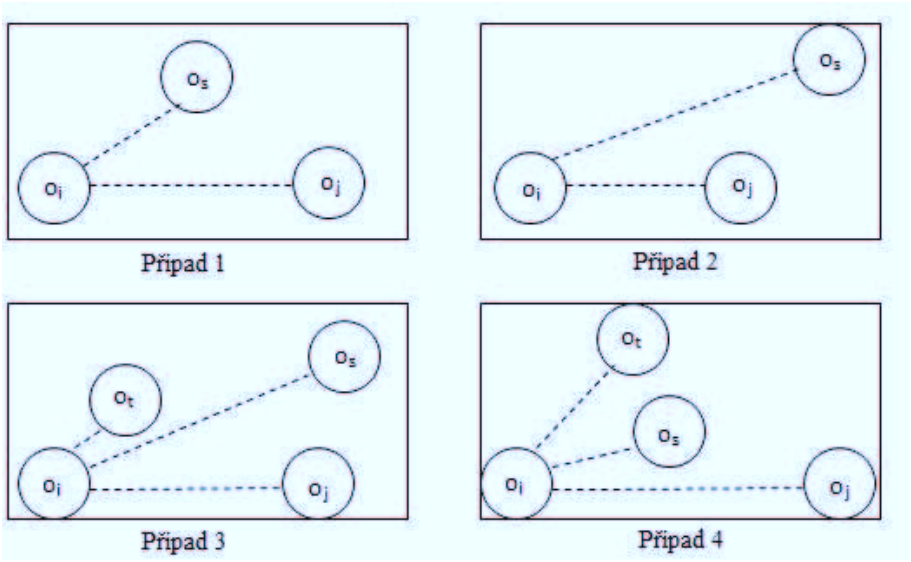
- **Situace:** o_i je objekt, o_s je nové centrum, které nahradí staré centrum o_j. $ndo_i = d_{ij}$ a $d_{ij} < d_{is}$.
- **Výsledek:** přepočítej vzdálenosti od všech center k o_i, a pak nastav nejkratší vzdálenost jako ndo_i.

Třetí případ

- **Situace:** o_i je objekt, o_j a o_t jsou staré středy shluku. o_s je nové centrum, který nahradil o_j. $ndo_i = d_{it}$ a $d_{it} < d_{is}$
- **Výsledek:** $ndo_i = d_{it}$. Hodnota ndo_i zůstává beze změny.

Čtvrtý případ

- **Situace:** o_i je objekt. o_j a o_t jsou staré centra shluku. o_s je nový střed shluku, který nahradil o_j. $ndo_i = d_{it}$ a $d_{it} > d_{is}$
- **Výsledek:** $ndo_i = d_{is}$



Obrázek 4: Případy přepočtu (převzato z [9])

4.5 Validace shluků

Validace shluků slouží ke kontrole kvality výsledných shluků. Prostřednictvím validace shluků můžeme provést výpočet jejich vlastností jako kulatost a kompaktnost. Dále můžeme např. s využitím vhodných metod vypočítat optimální počet shluků pro danou množinu dat. Podle typu přístupu můžeme rozdělit validace shluků na interní a externí.

4.5.1 Interní validace

U interní validace [14, 15] je vyhodnocení výsledných shluků založeno pouze na samotných shlucích, bez dalších informací. Validace se provádí několikrát pro různé nastavení algoritmu a vybírá se nejlepší výsledek shlukování. Technika je založena na předpokladu,

Algoritmus 8 OPC algoritmus

procedure OPC(počet shluků k, prahová hodnota ϵ, množina objektů O)
    Vytvoř matici vzdálenosti a podobnosti
    for (i=0;i<;i++) **do** (pro každý shluk)
        Náhodně vyber objekt, který ještě není centrum jako centrum pomocí CRF.
        Podle rovnice podobnosti a prahové hodnoty přiřaď objekty do shluku.
        Proveď update Mdv(o_i) a CRF(o_i) pro všechny o_i.
    end for
    Vypočítej cílovou hodnotu shluku.
    repeat(iterativně přizpůsobuj shluky)
        Dočasně nahraď jeden objekt středu shluku objektem, který není centrem shluku pomocí CRF podle algoritmu pro přepočet center (9).
        Vypočítej cílovou hodnotu nového shluku.
        Je-li nová cílová hodnota větší než maximální, pak ulož tuto novou cílovou hodnotu jako maximální cílovou hodnotu a ulož nové centrum shluku.
        Proveď update Mdv(o_i) and CRF(o_i) pro všechny o_i.
    until Do konvergence cílových hodnot
end procedure

Algoritmus 9 Přepočet starého středu za nový

procedure OPC(počet shluků k, prahová hodnota ϵ, množina objektů O)
    for (i = 1; i $\leq n$; i++) (pro každý objekt o_i) **do**
        if $ndo_i = d_{ij}$ **then**
            if $d_{is} \leq d_{ij}$ **then**
                n doi = dis(první případ)
            else
                for (r = 1; r $\leq$ k - 1; r++) (druhý případ) **do**
                    Vypočti vzdálenost z o_r do o_i
                end for
                Nastav nejkratší vzdálenost k ndo_i
            end if
        else if n$do_i > dis$ **then**
            $Ndo_i = dis$ (čtvrtý případ)
        else
            n do_i zůstane stejné (třetí případ)
        end if
    end for
end procedure

že členové shluku jsou blízko u sebe a daleko od členu jiných shluků. Slouží k výpočtu vlastnosti shluku jako je kulatost, kompaktnost aj.

- **Dunnův index**. Je definován jako poměr mezi minimální vzdálenosti dvou objektů z různých shluků, popř. center shluků a největší vzdálenosti dvou objektů v rámci shluku. Pro každý algoritmus se provádí vícekrát. Čím větší hodnota indexu, tím je výsledek shlukování kvalitnější. Vzdálenost mezi shluky a velikost shluku lze spočítat několika způsoby. Viz 3.2.3.

$$V(\varphi) = \frac{min_{i,j=1,..,k,i\neq j} d_C(C_i, C_j)}{max_{i=1,..,k} \Delta(C_k)}, kde : \tag{22}$$

  - $d_C(C_i, C_j)$ představuje minimální vzdálenost mezi dvěma shluky.
  - $\Delta(C_k)$ největší vzdálenost mezi dvěma objekty ve shluku.

- **Davies-Bouldinův index**. Cílem je identifikovat množiny shluků, které jsou kompaktní a dobře oddělené. Čím menší hodnota, tím lepší výsledek shlukování.

$$DB = \frac{1}{k} \sum_{i=1}^{k} max_{i\neq j} \frac{D_i + D_j}{d(c_i, c_j)}, kde : \tag{23}$$

  - k představuje počet shluků.
  - D_i, D_j představuje průměrnou vzdálenost objektu v danému shluku k centru daného shluku.
  - $d(c_i, c_j)$ představuje vzdálenost mezi shluky i a j,

- **Silhouett index**. Základem jsou tzv. siluety, které původně představovaly zobrazovací techniku vyhodnocující body ležící uvnitř a mimo shluk, založených na siluetě šířky daného bodu. Silueta šířky bodu představuje blízkost k vlastnímu shluku vzhledem k blízkosti jiných shluků. Pro daný bod se hodnota pohybuje od -1 do 1. Jestliže se hodnota bodu blíží k -1, znamená to, že bod je blíže k jinému shluku, než k tomu, ke kterému patří. Pokud se hodnota bodu blíží k 1, pak vzdálenost do vlastního shluku je výrazně menší než do jiného shluku. Čím je hodnota siluety vyšší, tím jsou shluky kompaktnější a lépe odděleny. Jedná se vlastně o indikátor členství daného objektu k danému shluku.

$$S(i) = \frac{(b(i) - a(i))}{max[a(i), b)(i)]}, kde : \tag{24}$$

  - a(i) představuje průměrnou vzdálenost mezi i-tým objektem a všemi objekty v daném shluku C_j.
  - b(i) představuje minimální vzdálenost mezi i-tým objektem a všemi objekty shluku $C_k, k \neq j$

  Výpočet a(i) a b(i) viz [14].

- **Další metody interní validace** [14], [15]:
  - Bic index
  - Calinski - Hrabasz index
  - Hubertova korelace s matici vzdálenosti

Následující validace shlukování jsou určeny pro soft shlukovací metody

- **Davies - Bouldin index pro Rough c-means**. Jedná se o klasický Davies - Bouldin index (23), který se liší pouze výpočtem průměrné vzdálenosti objektů ve shluku od centra tohoto shluku D_i. Viz [18].

$$D_j = \left\{ \begin{array}{ll} \omega_{low} \frac{\sum_{o_i \in \underline{R}(O_j)} \|o_i - c_j\|}{|\underline{R}(O_j)|} + \omega_{up} \frac{\sum_{o_i \in B(O_j)} \|o_i - c_j\|}{|B(O_j)|}, & \text{if } \underline{R}(O_j) \neq \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in B(O_j)} \|o_i - c_j\|}{|B(O_j)|}, & \text{if } \underline{R}(O_j) = \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in \underline{R}(O_j)} \|o_i - c_j\|}{|\underline{R}(O_j)|}, & \text{ostatní} \end{array} \right\} \tag{25}$$

- **Davies - Bouldin index pro Rough fuzzy c-means**. Stejný princip jako předcházející verze, lišící se opět pouze výpočtem průměrné vzdálenosti objektů od centra shluku rough fuzzy c-means. Viz [18].

$$D_j = \left\{ \begin{array}{ll} \omega_{low} \frac{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p \|o_i - c_j\|}{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p} + \omega_{up} \frac{\sum_{o_i \in B(O_j)} u_{ij}^p \|o_i - c_j\|}{\sum_{o_i \in B(O_j)} u_{ij}^p}, & \text{if } \underline{R}(O_j) \neq \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in B(O_j)} u_{ij}^p \|o_i - c_j\|}{\sum_{o_i \in B(O_j)} u_{ij}^p}, & \text{if } \underline{R}(O_j) = \emptyset, B(O_j) \neq \emptyset \\ \frac{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p \|o_i - c_j\|}{\sum_{o_i \in \underline{R}(O_j)} u_{ij}^p}, & \text{ostatní} \end{array} \right\} \tag{26}$$

- **Validace pro Fuzzy c-means**
  - **Partition koeficient**. Bezdekův index kvality shlukování udávající optimální rozdělení. Čím je index větší, tím je shlukování kvalitnější. Pro tento koeficient platí, že $\frac{1}{k} \leq PC \leq 1$, kde k značí počet shluků a u_{ij} příslušnost objektu o_i ke shluku c_k. [19]

$$PC = \frac{1}{k} \sum_{j=1}^{k} \sum_{i=1}^{m} u_{ij}^2 \tag{27}$$

  - **Partition entropy**. Bezdekův index kvality shlukování udávající optimální rozdělení. Čím je index menší, tím je shlukování kvalitnější. Pro tento koeficient platí, že $0 \leq PE \leq \log_2 k$, kde k značí počet shluků a u_{ij} příslušnost objektu o_i ke shluku c_g. [19]

$$PE = -\frac{1}{k} \sum_{j=1}^{k} \sum_{i=1}^{m} u_{ij} \log u_{ij} \tag{28}$$

– **Xie - Beni**. Jedná se o index zaměřený na dvě vlastnosti, a sice kompaktnost a separaci. Čitatel představuje kompaktnost fuzzy oddílu, jmenovatel sílu oddělení shluků. Čím je index menší, tím je výsledek shlukování lepší. [19]

$$XB = \frac{\sum_{j=1}^{k}\sum_{i=1}^{m} u_{ij}^2 \|o_i - c_j\|}{mmin\|c_j - c_i\|}, kde: \tag{29}$$

4.5.2 Externí validace

U externí validace [14, 15] se provádí kontrola výsledných shluků vzniklých daným algoritmem s externími informacemi, které udávají jak by výsledné shluky měly vypadat.

- **Randův index, Jaccardův koeficient, Folkes-Mallowsův index**. Základem této metody jsou dvě množiny. V obou těchto množinách je rozděleno m objektů do k skupin. Množina rozdělení do shluků podle daného algoritmu C^A a množina správného rozdělení (externí informace) C^B. Pro každé dva objekty o_i a o_j, kde $o_i \neq o_j$ mohou nastat čtyři případy:
  - o_i a o_j patří do stejného shluku v C^A i v C^B.
  - o_i a o_j patří do stejného shluku v C^A, ale do různých shluků v C^B.
  - o_i a o_j patří do různých shluků v C^A, ale do stejného v C^B.
  - o_i a o_j patří do různých shluků v C^A i v C^B.

  Pro každý z těchto případů se provede celkový součet pro všechny dvojice objektů, pro které nastal. Ty se označí jako a, b, c, d (a = první případ atd.) a M = m(m-1)/2 je počet všech různých dvojic.

  Randův index:

$$R = \frac{a+d}{M} \tag{30}$$

  Jaccardův koeficient:

$$J = \frac{a}{a+b+c} \tag{31}$$

  Folkes-Mallowsův index:

$$FM = \sqrt{\frac{a}{a+b}\frac{a}{a+c}} \tag{32}$$

- **Hubertova korelace** Stejně jako u předchozích metod i zde se pracuje se dvěma množinami C^A a C^B. Pro každou množinu C, je vztah mezi dvěma objekty určující, zda patří do stejného shluku zastoupen matici podobnosti s(i,j), kde s(i,j) = 1, když o_i i o_j patří do stejného shluku a s(i,j) = 0, když patří do různých shluků.

$$\Gamma = \frac{1}{M}\sum_{i=1}^{m-1}\sum_{j=i+1}^{m} s^A(i,j)s^B(i,j), kde: \tag{33}$$

  - M = m(m-1)/2

- **Další metody externí validace** [15]:
  - F-measure
  - Purity
  - NMI
  - Entropie

5 Implementace vybraných algoritmů

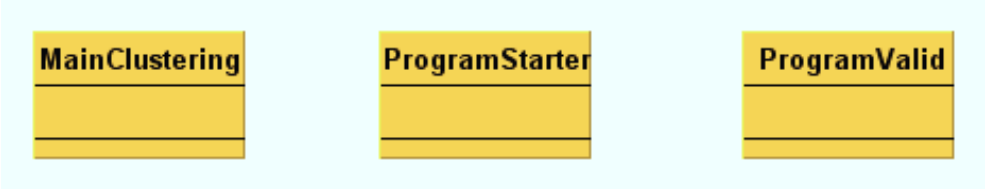
Pro implementaci jsme zvolili programovací jazyk C# v .NET frameworku 4.0. Implementace vybraných algoritmů (k-means, fuzzy c-means, rough c-means a rough fuzzy c-means) byla rozdělena do několika knihoven, jejichž popis následuje.

Celá aplikace je rozdělana do tří knihoven, a sice:

- Clustering
- ClusteringMethod
- Utility

5.1 Knihovna Clustering

Tato knihovna je vstupním bodem celého programu, řídí její další běh na základě zvolené funkcionality uživatele, validuje tyto parametry zadané uživatelem. Podle zvolených parametrů pak volí další funkcionalitu.



Obrázek 5: Zjednodušený třídní diagram knihovny Clustering

- **MainClustering** - Vstupní bod aplikace.
- **ProgramStarter** - Podle zadaných parametrů z řídí další běh programu.
- **ProgramValid** - Kontroluje parametry zadané uživatelem, popř. provede přerušení chodu programu.

5.2 Knihovna ClusteringMethod

Knihovna obsahující třídy pro vykonání daného shlukovacího algoritmu a další podpůrné třídy k jejich chodu.

- **AbstractDistance** - Bázová třída pro výpočet vzdáleností.
- **Euclidean** - Třída pro výpočet euklidovské vzdálenosti mezi dvěma objekty, respektive objektem a centrem shluku.
- **Manhattan** - Třída pro výpočet manhattanské vzdálenosti mezi dvěma objekty, respektive objektem a centrem shluku.

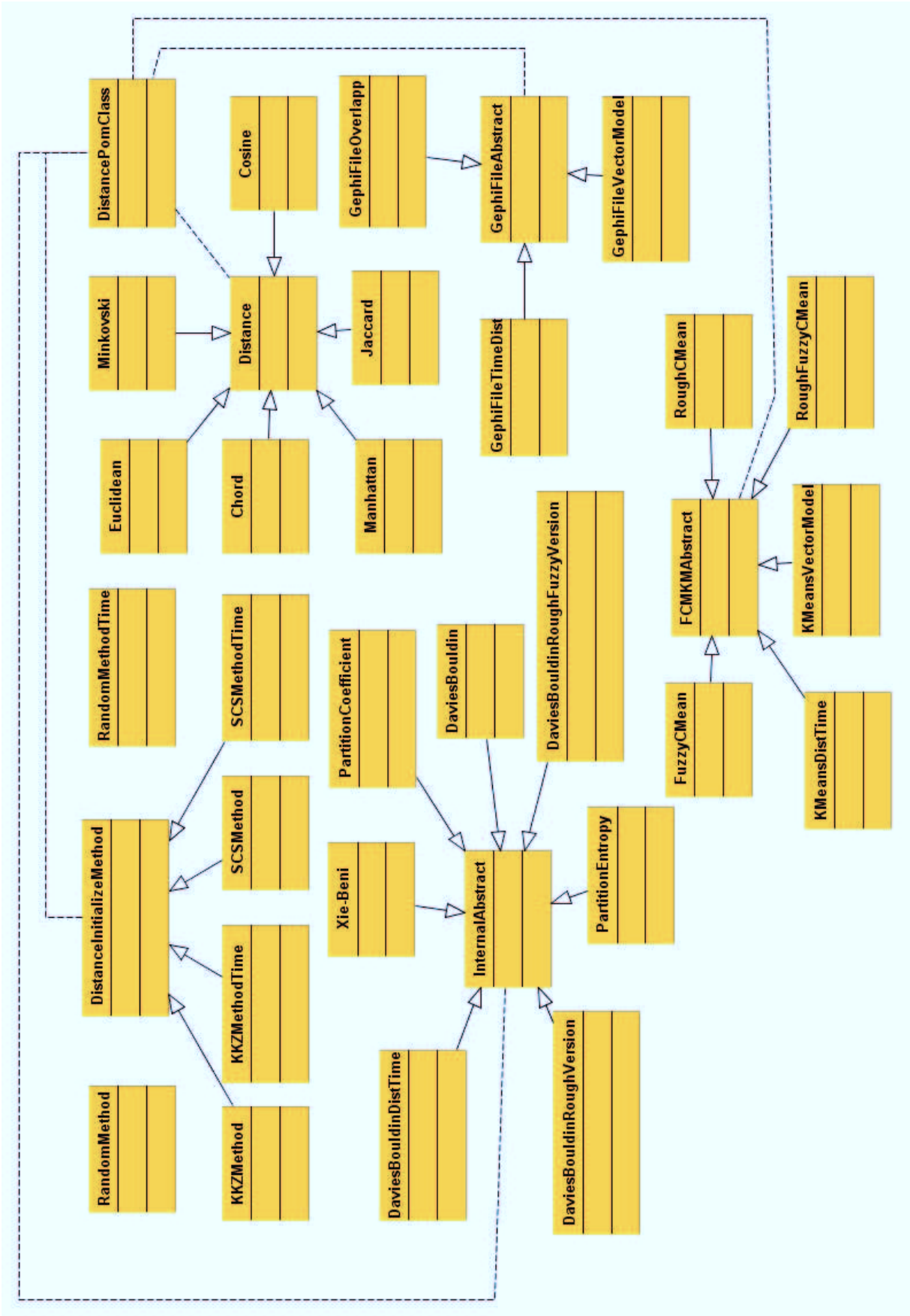
- **Chord** - Třída pro výpočet tětivové vzdálenosti mezi dvěma objekty, respektive objektem a centrem shluku.
- **Cosine** - Třída pro výpočet kosinové vzdálenosti z kosinové podobnosti mezi dvěma objekty, respektive objektem a centrem shluku.
- **Jaccard** - Třída pro výpočet jaccardovy vzdálenosti z jaccardovy podobnosti mezi dvěma objekty, respektive objektem a centrem shluku.
- **DistanceInitializeMethod** - Bázová třída pro generování center shluků.
- **KKZMethod** - Třída pro generování center metodou KKZ.
- **SCSMethod** - Třída pro generování center metodou SCS.
- **KKZMethodTime** - Třída pro generování center metodou KKZ, pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků.
- **SCSMethodTime** - Třída pro generování center metodou SCS, pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků.
- **RandomMethod** - Třída pro náhodné generování center shluků.
- **RandomMethodTime** - Třída pro náhodné generování center shluků, pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků.
- **FCMKMAbstract** - Bázová třída pro jednotlivé shlukovací algoritmy.
- **KMeansVectorModel** - Třída implementující algoritmus K-means.
- **KMeansDistTime** - Třída implementující algoritmus K-means pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků.
- **FuzzyCMean** - Třída implementující algoritmus Fuzzy C-means.
- **RoughCMean** - Třída implementující algoritmus Rough C-means.
- **RoughFuzzyCMean** - Třída implementující algoritmus Rough Fuzzy C-means.
- **InternalAbstract** - Bázová třída pro validaci výsledků shlukování.
- **DaviesBouldin** - Třída implementující Davies Bouldinův index.
- **DaviesBouldinDistTime** - Třída implementující Davies Bouldinův index pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků.
- **DaviesBouldinRoughVersion** - Třída implementující Davies Bouldinův index pro Rough C-means.
- **DaviesBouldinRoughFuzzyVersion** - Třída implementující Davies Bouldinův index pro Rough Fuzzy C-means.

- **PartitionCoefficient** - Třída implementující Partition koeficient pro Fuzzy C-means.
- **PartitionEntropy** - Třída implementující Partition entropy pro Fuzzy C-means.
- **Xie-Beni** - Třída implementující Xie - Beni index pro Fuzzy C-means.
- **GephiFileAbstract** - Bázová třída pro jednotlivé generování souborů pro Gephi.
- **GephiFileVectorModel** - Třída implementující generátor souboru pro Gephi, pro algoritmus K-means.
- **GephiFileTimeDist** - Třída implementující generátor souboru pro Gephi, pro algoritmus K-means, pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků.
- **GephiFileOverlapp** - Třída implementující generátor souboru pro Gephi, pro soft shlukovací algoritmy.

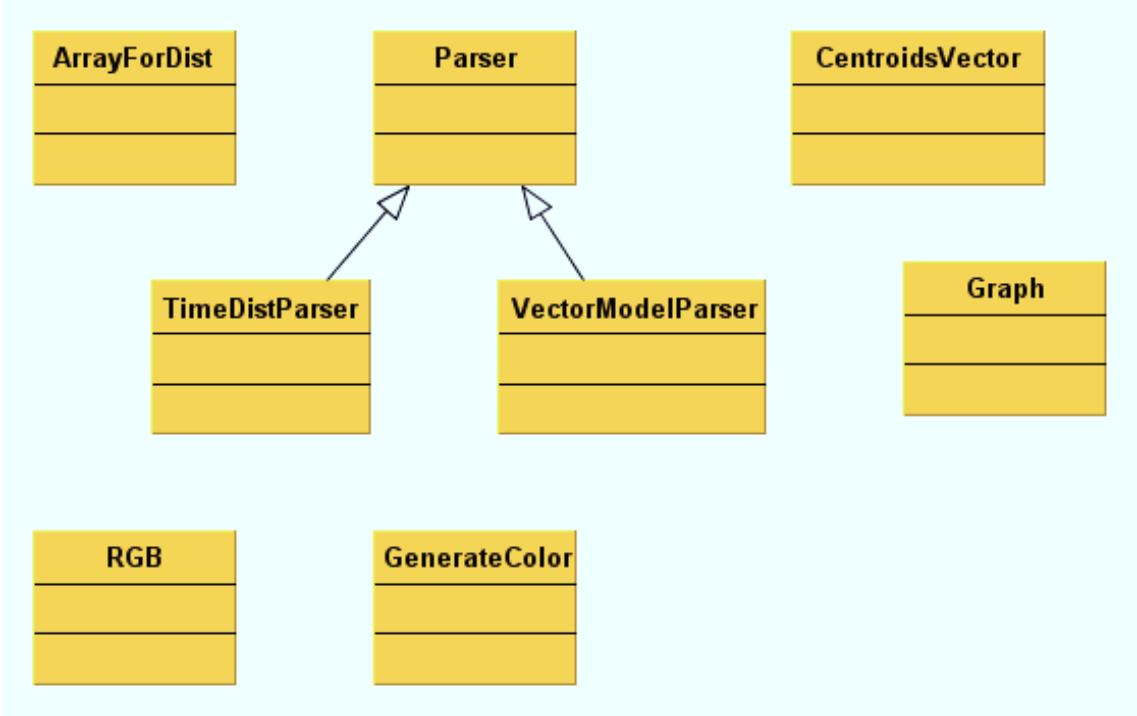
5.3 Knihovna Utility

Obsahuje pomocné třídy pro podporu shlukovacích algoritmů a třídy pro zpracování dat do podoby vhodné pro vstup algoritmů.

- **ArrayForDist** - Pomocná třída pro vytvoření pole, pro výpočet vzdálenosti mezi objekty, respektive centry a objekty.
- **RGB** - Pomocná třída používána třídou GenerateColor
- **GenerateColor** - Třída pro generování barev uzlů pro Gephi soubor.
- **Graph** - Pomocná třída pro generování hran uzlů a jejich vah a pomocná třída pro indexy sloužící k validaci shluků.
- **CentroidsVector** - Pomocná třída pro generovaní center shluků.
- **Parser** - Bázová třída pro načtení dat vektorového modelu z textového souboru a pro kontrolu správnosti těchto dat.
- **TimeDistParser** - Třída implementující načtení dat pro algoritmus K-means, pro vektorový model, kde je atribut průměrný čas odeslaných příspěvků a pro kontrolu správnosti těchto dat.
- **VectorModelParser** - Třída implementující načtení pro vektorové modely a pro kontrolu správnosti těchto dat.



Obrázek 6: Zjednodušený třídní diagram knihovny ClusteringMethod



Obrázek 7: Zjednodušený třídní diagram knihovny Utility

6 Experimenty a vizualizace shlukování

V této kapitole jsou uvedeny výsledky experimentů prováděných s některými z algoritmů uvedených v kapitole (4) a vizualizace shluků provedené v nástroji Gephi na základě rozdělení objektů prostřednictvím jednotlivých algoritmů. Cílem bylo zjistit, jak budou shluky vypadat při změně jejich vlastností a vstupních nastavení, jako např. počet shluků, do kterých se mají objekty shlukovat, změnou metod výpočtu vzdáleností a dalších možných nastavitelných parametrů aj.

U algoritmu K-means je provedena validace při shlukování do 2, 4, 8, 10, 15, 20 shluků, u soft shlukovacích algoritmů při shlukování do 2, 4, 8, a 10 shluků s využitím metody KKZ (4.2.1) pro inicializaci center. Jako vstupní množina dat jsou použity vektorové modely získané z extrahovaných dat, viz. kapitola (3.1). Jako metriky vzdálenosti byly u všech metod zvoleny euklidovská vzdálenost a vzdálenost vycházející z kosinové podobnosti (3.2.3). Další volené parametry jsou uvedeny přímo u daného algoritmu.

Experimenty byly prováděny na algoritmech rodiny K-means, tedy samotném algoritmu K-means a dále pak na algoritmech Fuzzy c-means a Rough c-means. Všechny výsledky algoritmů jsou validovány prostřednictvím Davies-Bouldinova a Dunnova indexu u algorimtu K-means, Fuzzy c-means pomocí Xie-Beni indexu a partition entropy a koeficientu. Rough c-means je validován upravenou verzí Davies-Bouldinova indexu pro tento algoritmus.

Vizualizace byla provedena v nástroji **Gephi**, což je profesionální open source nástroj pro vizualizaci a analýzu sítí a grafů podporující celou řadu typů souborů.

Po vytvoření souboru pro Gephi a jeho nahrání do tohoto vizualizačního nástroje se prostřednictvím vybraného layoutu začne „tvarovat“ výsledný graf. Gephi obsahuje shlukovací algoritmus, který provede rozdělení objektů do shluku na základě ohodnocení hran (podobnosti objektů), což nám může sloužit jako pomůcka pro ověření kvality shluků.

Generování souboru pro Gephi, pro k-means a soft shlukovací

- **Generování souboru pro Gephi, pro k-means**. Jako zdroj dat byly použity vektorové modely diskuzního fóra, viz. (3.1). Na základě těchto vstupních dat, byl při aplikaci algoritmu k-means, který provede rozdělení objektů do shluků a na základě matice podobnosti, vygenerován soubor ***.gml** pro vizualizační nástroj Gephi.
  Popis kroků pro vygenerování souboru *.gml:
  - Prostřednictvím k-means algoritmu proveď rozdělení m objektů do k shluků.
  - Vygeneruj k barev pro k shluků.
  - Projdi každé dvě dvojice objektů.
  - Jestliže patří do stejného shluku, vypočítej ohodnocení hrany mezi objekty a jestliže je toto ohodnocení (podobnost mezi objekty) větší jak zadaná prahová hodnota, pak ulož objekty, pokud nejsou uloženy a ulož hodnotu hrany.

  - Jestliže patří do různých shluků, vypočítej ohodnocení hrany mezi objekty a jestliže je toto ohodnocení (podobnost mezi objekty) větší jak zadaná prahová hodnota, pak ulož objekty, pokud nejsou uloženy a ulož hodnotu hrany.

  Jelikož je vstupní množina dat celkem rozsáhlá, je třeba použit takovou prahovou hodnotu, aby výsledný soubor nebyl příliš velký, příliš malý, popř. neobsahoval mnoho odlehlých hodnot. Nastavení prahových hodnot je uvedeno u každé testované množiny dat.

- **Generování souboru pro Gephi, pro soft shlukovací alforitmy**. Pro vizualizaci výsledku překrývajícího se shlukování je opět generován *gml soubor pro vizualizační nástroj Gephi. To je prováděno následujícím způsobem:
  - Pro každý shluk byl vytvořen jeden uzel, jež obsahuje počet objektů, které výhradně patří k tomuto shluku.
  - Hrana mezi dvěma uzly udává počet objektů, které mají tyto shluky společné.
  - Jestliže nemají shluky společné žádné objekty, pak hrana mezi těmito uzly neexistuje.

Gml soubor

```
graph [
    node [
    id  0
    label  "Shluk␣c.0␣–␣1"
        graphics [
            fill   "#52B56C"
        ]
    ]
    node [
    id  1
    label  "Shluk␣c.1␣–␣1"
        graphics [
            fill   "#16C399"
        ]
    ]
    edge [
    id  1
        source 0
    target  1
        label  "98"
    ]
]
```

Výpis 3: Ukázka struktury gml souboru

Při generování výsledků shlukování jsou kromě gml souboru pro vizualizační nástroj Gephi, generovány dva textové soubory, a sice soubor obsahující zvolená centra a soubor s výsledky shlukování, jež obsahuje:

- Seznam objektů a jejich příslušnost k jednotlivým shlukům.

- Počet objektů, které patří k více shlukům a jejich seznam.
- Počet objektů patřících do jednotlivých shluků.
- Hodnota daných validačních indexů pro daný algoritmus.

6.1 Experimenty s k-means

Pro vizualizaci shlukování získaných dat bylo využito algoritmu K-means. Jak je již uvedeno v kapitole (4), nejedná se přímo o soft shlukovací algoritmus, ale i přesto bude na něm možno demonstrovat překrývající se shlukování v případě, že nebudou vykreslovány hrany objektů pouze mezi objekty, které patří do stejného shluku, ale i hrany mezi objekty z rozdílných shluků.

Jestliže bychom nevykreslovali hrany mezi objekty z rozdílných shluků, provedl by Gephi shlukování objeků do separatních shluků. Tím docílíme toho, že některé shluky se budou překrývat, respektive objekty shluků. Přidáním hran mezi objekty z rozdílných shluků provede Gephi shlukování tak, že některé objekty z jednoho shluku, zasahují do shluku jiného a díky tomu vypadá výsledný graf složený ze shuků, jako překrývající se.

6.1.1 Experimenty s vektorovým modelem, kde každý atribut představuje počet příspěvků daného uživatele v dané diskuzi.

Vzhledem k tomu, že jednotlivé vektory vektorového modelu mají velký počet nulových hodnot a počet jejich atributu je velmi nízký a některé vektory mají zase i několik tisíců atributu a ty se při výpočtu vzdálenosti musí porovnávat, je toto výpočetně náročné. Proto tento vektorový model pro testování je nevhodný a je testován pouze pro prvních 100 objektů vektorového modelu.

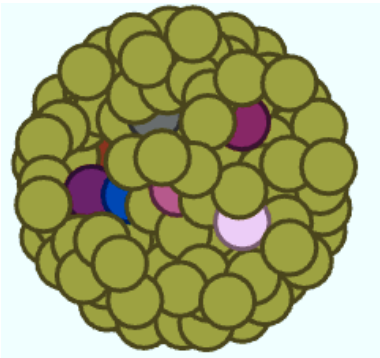
Nejlepších výsledků, tabulka 6, bylo dosaženo při shlukování do dvou shluků, a to jak při použití euklidovské míry, tak i při použití míry vycházející z kosinové podobnosti.

Na níže uvedených obrázcích 8 a 9 je zobrazeno shlukování pro obě míry do 10 shluků. U euklidovské míry lze vidět, že shlukování objektů je nerovnoměrné, většina objektů patří do jednoho shluku, a to ke shluku, jehož centrum má nejmenší počet atributů nebo takové atributy, jež jsou nejvíce rozdílné s atributy ostatních objektů. Díky tomu je vzdálenost mezi objekty menší a výsledná podobnost větší. Je to dáno také tím, že celkový počet atributů objektů je přes 30 000 a jen malý zlomek objektů má společné atributy. Z toho je patrné, že použití euklidovské vzdálenosti je pro taková data nevhodné. Tabulka 7, kde c_j představuje číslo shluku, obsahuje počet objektů, který obsahuje příslušný shluk.

U druhé použité metriky vycházející z kosinové podobnosti je rozložení objektů do shluku o něco lepší, jelikož zohledňuje podobnost mezi objekty a jejich atributy. Ovšem jak je vidět z tabulky 8 i zde není rozložení objektů do ideální, což je dáno nevhodnou množinou vstupních dat.

Euklidovská míra		
Počet shluků	**Dunn index**	**Davies - Bouldin index**
2	**1.74**	**0.045**
4	0.766	0.079
8	1.08	0.061
10	1.23	0.065
15	0.53	0.11
20	0.31	0.25
Míra vycházející z kosinové podobnosti		
Počet shluku	**Dunn index**	**Davies - Bouldin index**
2	**0.99**	**0.093**
4	0.88	1.49
8	0.87	1.25
10	0.87	1.19
15	0.71	1.19
20	0.71	1.12

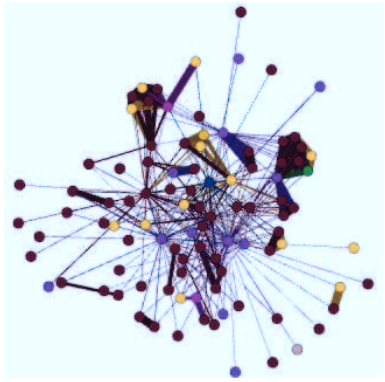
Tabulka 6: K-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi.**



Obrázek 8: K-means, euklidovská vzdálenost, 10 shluků, prahová hodnota mezi vrcholy patřící do stejného shluku - 0.001, prahová hodnota mezi vrcholy různých shluků shluku - 0.001

6.1.2 Experimenty s vektorovým modelem, kde každý atribut představuje počet příspěvků daného uživatele odeslaného v daném časovém rozmezí

Oproti předchozímu vektorovému modelu, se zde pracuje s celou množinou dat a i zde bylo dosaženo nejlepších výsledku při shlukování s využitím u euklidovské vzdálenosti do dvou shluků. U metriky vycházející z kosinové podobnosti byl výsledek pro Dunn index nejlepší u shlukování do dvou shluku, u Davies - Bouldin indexu při shlukování do 20ti shluků, viz tabulka 9.



Obrázek 9: K-means, vzdálenost vycházející z kosinové podobnosti, 10 shluků, prahová hodnota mezi vrcholy patřící do stejného shluku - 0.001, prahová hodnota mezi vrcholy různých shluků shluku - 0.001

	Euklidovská míra									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	99	-	-	-	-	-	-	-	-
4	1	1	1	97	-	-	-	-	-	-
8	1	1	1	1	1	1	1	93	-	-
10	1	1	1	1	1	1	1	1	1	91
15	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1
k shluků	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}
2	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
15	1	1	1	1	86	-	-	-	-	-
20	1	1	1	1	1	1	1	1	2	80

Tabulka 7: K-means, počet objektů patřících do daného shluku - euklidovská vzdálenost

Na obrázcích 10 a 11 jsou vidět výsledky shlukování v nástroji Gephi. Obrázek 10 představuje vizualizaci shlukování při použití euklidovské metriky. Stejně jako u předcházejícího vektorového modelu, i zde je vidět, že tato míra vzdálenosti není vhodná, jelikož dojde ke shlukování většiny objektů k jednomu shluku, viz tabulka 10.

Obrázek 11 představuje vizualizaci při využití vzdálenosti z kosinové podobnosti. Použití této metriky je vhodné, objekty jsou rovnoměrněji rozloženy do jednotlivých shluků, viz tabulka 11.

Pozn. Vzhledem k tomu, že vektorový model obsahuje přibližně 9000 objektů, je zde

	Metrika vycházející z kosinové podobnosti									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	99	1	-	-	-	-	-	-	-	-
4	72	1	4	23	-	-	-	-	-	-
8	70	1	4	13	1	9	1	1	-	-
10	68	1	4	13	1	9	1	1	1	1
15	52	1	4	13	1	7	1	1	1	1
20	51	1	4	10	1	6	1	1	1	1
k shluků	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}
2	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
15	2	1	5	2	8	-	-	-	-	-
20	2	1	2	2	8	1	1	1	1	1

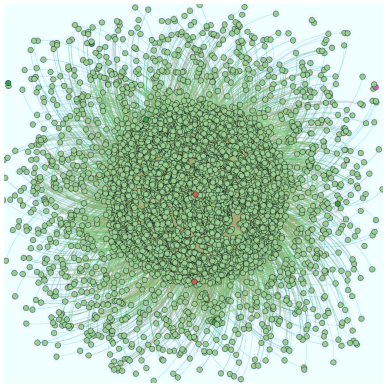
Tabulka 8: K-means, počet objektů patřících do daného shluku - vzdálenost vycházející z kosinové podobnosti

Euklidovská míra		
Počet shluků	**Dunn index**	**Davies - Bouldin index**
2	**2.16**	**0.003**
4	0.49	0.65
8	0.29	0.67
10	0.45	0.67
15	0.28	0.57
20	0.34	0.49
Míra vycházející z kosinové podobnosti		
Pocet shluku	**Dunn index**	**Davies - Bouldin index**
2	**0.28**	3.55
4	0.13	4.82
8	0.15	2.63
10	0.146	2.88
15	0.15	1.88
20	0.145	**1.72**

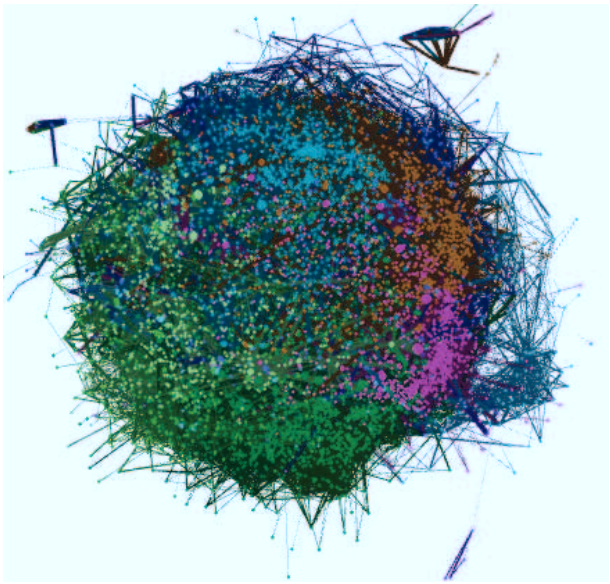
Tabulka 9: K-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**

nutno nastavit větší prahovou hodnotu pro uložení dat do souboru, k vizualizaci v nástroji Gephi, jelikož výstupní soubor by mohl mít až 9000 uzlů a $\frac{9000^2}{2}$ hran. Proto se

zde volí vhodná prahová hodnota, aby výstupní soubor nebyl příliš veliký a generování souboru výpočetně náročné. Na druhé straně je volit takovou hodnotu, aby výsledný graf neměl příliš odlehlých hodnot. Proto je dobrá znalost vstupních dat a podle toho volit prahovou hodnotu.



Obrázek 10: K-means, euklidovská vzdálenost, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, 10 shluků, prahová hodnota mezi vrcholy patřící do stejného shluku - 0.9998, prahová hodnota mezi vrcholy různých shluků shluku - 0.999



Obrázek 11: K-means, vzdálenost vycházející z kosinové podobnosti, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, 10 shluků, prahová hodnota mezi vrcholy patřící do stejného shluku - 0.7, prahová hodnota mezi vrcholy různých shluků shluku - 0.999

	Euklidovská míra									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	8922	-	-	-	-	-	-	-	-
4	1	8893	26	3	-	-	-	-	-	-
8	1	8859	1	1	15	4	40	2	-	-
10	1	8859	1	1	3	1	2	2	40	13
15	1	8810	1	1	1	1	1	1	18	2
20	1	8810	1	1	1	1	1	1	1	1
k shluků	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}
2	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
15	1	5	1	71	8	-	-	-	-	-
20	1	3	1	18	1	71	3	1	1	4

Tabulka 10: K-means, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, počet objektů patřících do daného shluku - euklidovská vzdálenost

	Metrika vycházející z kosinové podobnosti									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	5457	3466	-	-	-	-	-	-	-	-
4	3148	2279	1155	2341	-	-	-	-	-	-
8	646	1894	972	449	1694	854	873	1541	-	-
10	1230	509	283	46	1707	897	415	1303	827	1706
15	553	220	757	264	43	772	93	1319	151	665
20	525	208	91	16	25	728	84	694	145	455
k shluků	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}
2	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
15	466	995	564	671	1390	-	-	-	-	-
20	259	601	502	409	659	394	389	788	762	1189

Tabulka 11: K-means, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, počet objektů patřících do daného shluku - Metrika vycházející z kosinové podobnosti

6.1.3 Vektorový model s průměrným časem odeslaných příspěvku

V tomto případě není použita žádná míra pro výpočet vzdálenosti objektů, ale vzhledem k tomu, že každý objekt obsahuje pouze jeden atribut, a sice průměrnou dobu, kdy uživatel odesílá příspěvky, vzdálenost mezi těmito objekty je vypočítána jako absolutní hodnota rozdílu dvou objektů.

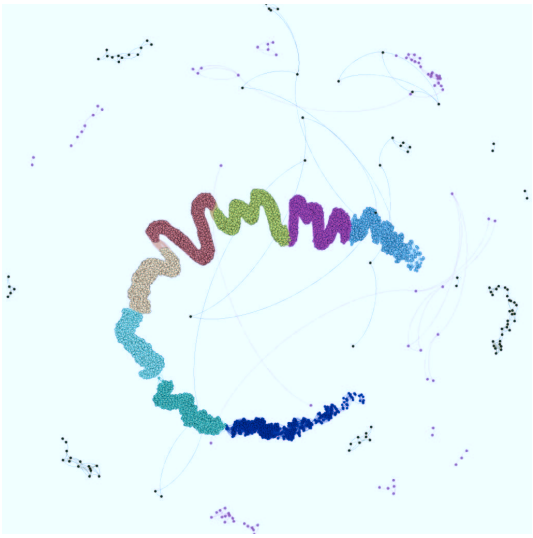
Podle použitých indexu pro validaci shlukování bylo nejlepších výsledku dosaženo při shlukování do 15 shluků (Davies - Bouldin index), respektive do 20 shluků (Dunn index), viz tabulka 12.

Euklidovská míra		
Počet shluků	**Dunn index**	**Davies - Bouldin index**
2	0.37	0.63
4	0.44	0.535
8	0.37	0.51
10	0.35	0.5
15	0.375	**0.495**
20	**0.45**	0.496

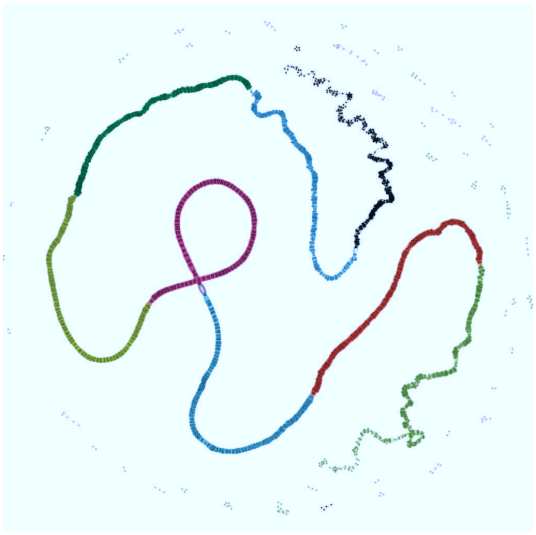
Tabulka 12: K-means, validace shlukování pro vektorový model s **průměrným časem odeslaných příspěvku**

Na obrázcích 12 a 13 je vidět, že při použití různě velkých prahových hodnot může být výsledná podoba shluků rozdílná. Dále si je možné všimnout odlehlých objektů kolem grafu, to je dáno opět zvolenou prahovou hodnotou. Kdybychom zvolili nižší prahové hodnoty, odlehlých hodnot by bylo sice méně, ovšem výsledný soubor pro vizualizaci by byl velký a těžko zpracovatelný vizualizačním nástrojem Gephi.

Tabulka 13 obsahuje počet objektů, které obsahuji jednotlivé shluky při různém počtu shluků.



Obrázek 12: K-means, vektorový model s **průměrným časem odeslaných příspěvku**, 10 shluků, prahová hodnota mezi vrcholy patřící do stejného shluku - 0.997, prahová hodnota mezi vrcholy různých shluků shluku - 0.999



Obrázek 13: K-means, vektorový model s **průměrným časem odeslaných příspěvku**, 10 shluků, prahová hodnota mezi vrcholy patřící do stejného shluku - 0.998, prahová hodnota mezi vrcholy různých shluků shluku - 0.999

6.2 Experimenty z Fuzzy C-means

U Fuzzy C-means a u dalších překrývajících se algoritmů je prováděno testováni pouze na prvních dvou vektorových modelech, a sice vektorovém modelu, kde každý atribut

k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	5792	3122	-	-	-	-	-	-	-	-
4	2687	192	1950	4085	-	-	-	-	-	-
8	972	92	1195	2141	791	1679	1815	229	-	-
10	720	81	970	1673	417	1242	1383	104	675	1649
15	376	58	585	1340	78	879	975	34	323	1172
20	279	43	484	958	39	552	711	15	210	862
k shluků	c_{10}	c_{11}	c_{12}	c_{13}	c_{14}	c_{15}	c_{16}	c_{17}	c_{18}	c_{19}
2	-	-	-	-	-	-	-	-	-	-
4	-	-	-	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-
15	1129	161	609	737	458	-	-	-	-	-
20	840	73	417	541	328	21	17	658	910	956

Tabulka 13: K-means, vektorový model s **průměrným časem odeslaných příspěvku**, počet objektů patřících do daného shluku

představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí** a vektorovém modelu, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi.**

K testování byly použity, stejně jako pro všechny zbylé překrývající se algoritmy euklidovská vzdálenost a vzdálenost vycházející z kosinové podobnosti, pro fuzziness koeficient (koeficient míry překryvu) 1.2 a 2.0 při shlukování do 2, 4, 8 a 10ti shluků. Prahová hodnota byla zvolena 0.1, tzn. že jestliže bude mít objekt příslušnost k nějakému shluku větší jak 0.1, pak bude tohoto shluku patřit.

Pro validaci bylo použito partition koeficientu, partition entropy a Xie - Beni indexu.

6.2.1 Experimenty s vektorovým modelem, kde každý atribut představuje počet příspěvků daného uživatele odeslaného v dané diskuzi

Nejlepších výsledku shlukování bylo dosaženo pro shlukování do dvou shluků při využití euklidovské vzdálenosti a fuzziness koeficient (míra překryvu shluků) 1.2, tab. 14 , a to pro všechny indexy pro validaci shluků. U partition keoficientu byly výsledky pro shlukování do různého počtu shluku takměř stejné. U druhe metriky vycházející z kosinové podobnosti, tab. 14 byly výsledky stejné, tedy nejlepšího výsledku bylo dosaženo opět pro dva shluky s výjimkou xie - beni indexu, zde bylo dosaženo nejlepšího výsledku při shlukování do 8 shluků.

Při nastavení fuzziness koeficientu na 2.0, tab. 15, byly výsledky shlukování u všech indexů kvality shlukování i obou typů metriky výsledky stejné, a sice pro shlukování do dvou shluků.

Na obrázku 15 a 15 lze vidět příslušnost objektu ke shlukům. Při použití nejmenšího fuzziness koeficientu 1.2 nedochází k žádnému překryvu mezi shluky (neexistují hrany

Euklidovská vzdálenost, fuzziness koeficient - 1.2			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.99**	**0.00003**	**0.045**
4	0.98	0.04	0.18
8	**0.99**	0.0012	0.078
10	**0.99**	0.0004	0.098
Míra vycházející z kosinové podobnosti, fuzziness koeficient - 1.2			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.56**	**0.896**	0.59
4	0.39	1.635	0.34
8	0.286	2.446	**0.19**
10	0.39	2.25	0.195

Tabulka 14: Fuzzy c-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, fuzziness - 1.2

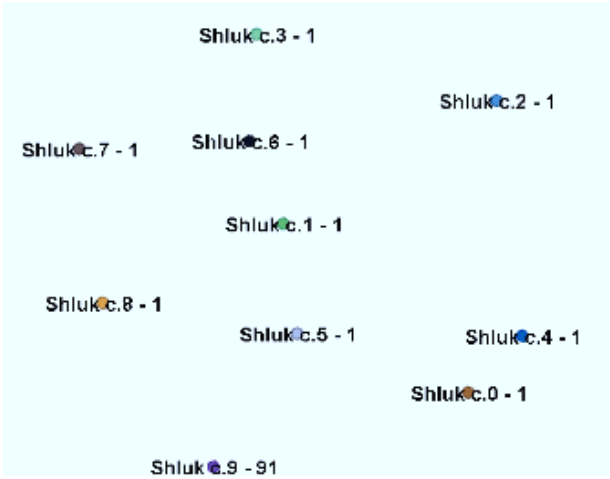
Euklidovská míra, fuzziness koeficient - 2.0			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.935**	**0.16**	**0.15**
4	0.48	1.12	6.83
8	0.26	2.09	376557270
10	0.22	2.31	21146
Míra vycházející z kosinové podobnosti, fuzziness koeficient - 2.0			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.51**	**0.98**	**0.605**
4	0.265	1.96	80.1
8	0.127	2.99	3648.62
10	0.1	3.31	7854.4

Tabulka 15: Fuzzy c-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, fuzziness 2.0

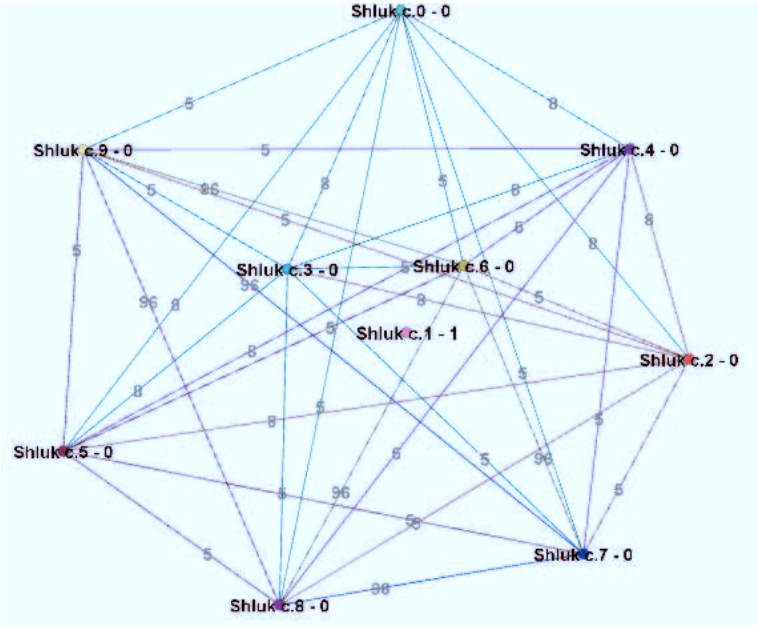
mezi jednotlivými uzly - shluky). Při použití maximálního fuzziness koeficientu 2.0 dochází k velkému překryvu mezi shluky. Objekt, který patří pouze do jednoho shluku, patří pouze do shluku č.1, tento shluk se zároveň nepřekrývá s žádným dalším shlukem, neexistuje hrana s jiným uzlem. U ostatních uzlu neexistuje ve shluku objekt, která by patřil pouze do daného shluku. Hrany s ostatními shluky naznačují velkou míru překryvu.

V tabulkách 16 a 17 je uvedeno kolik objektů patři pouze do daného shluku při shlukování do různého počtu shluků a kolik objektů patří do daného shluku, i když mohou být objekty tohoto shluku součástí shluku jiného.

Na obrázcích 16 a 17 lze vidět, že při použití vzdálenosti vycházející z kosinové podobnosti jsou výsledky při použití fuzziness koeficientu 1.2 dochází k většímu překryvu než



Obrázek 14: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, fuzziness - 1.2, euklidovská vzdálenost



Obrázek 15: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, fuzziness - 2.0, euklidovská vzdálenost

u předcházející míry. Při použití fuzziness koeficientu 2.0 k rovnoměrnějšímu překryvu mezi všemi shluky, viz tabulka 19. U předchozí míry docházelo k výrazným překryvům pouze u poloviny shluků, viz tabulka 17.

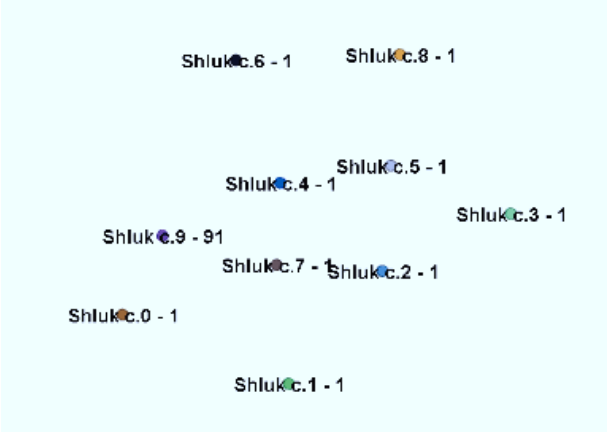
	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	99	1	-	-	-	-	-	-	-	-
4	1	1	0	93	-	-	-	-	-	-
8	1	1	1	1	1	1	1	91	-	-
10	1	1	1	1	1	1	1	1	1	91
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	99	1	-	-	-	-	-	-	-	-
4	1	1	5	98	-	-	-	-	-	-
8	1	1	1	1	1	1	1	93	-	-
10	1	1	1	1	1	1	1	1	1	91

Tabulka 16: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a fuzziness koef. 1.2

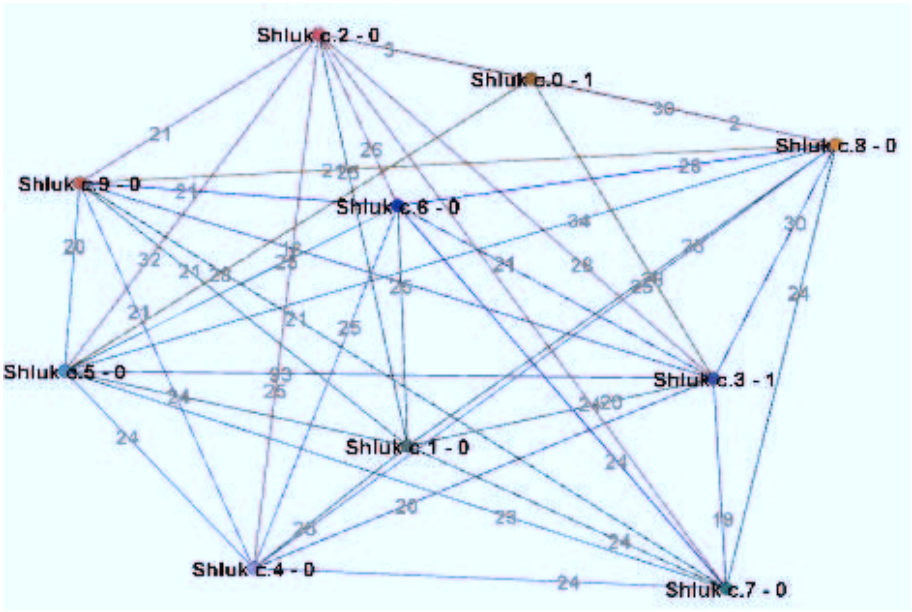
	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	0	88	-	-	-	-	-	-	-	-
4	1	1	0	0	-	-	-	-	-	-
8	1	1	0	0	0	0	0	0	-	-
10	1	0	0	1	0	0	0	0	0	0
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	12	100	-	-	-	-	-	-	-	-
4	1	14	99	99	-	-	-	-	-	-
8	1	1	8	8	98	98	98	98	-	-
10	8	1	8	8	8	8	96	96	96	96

Tabulka 17: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a fuzziness koef. 2.0

Pozn.: Kromě toho vlivu použitého fuzziness koeficientu a použité míry vzdálenosti míru překryvu, objekty spadajících do více shluků, můžeme ovlivnit použitím vyšší prahové hodnoty. Tzn. jestliže objekt nebude mít k danému shluku, alespoň takovou příslušnost jako je prahová hodnota, nebude do tohoto shluku patřit.



Obrázek 16: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, fuzziness - 1.2, vzdálenost vycházející z kosinové podobnosti



Obrázek 17: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, fuzziness - 2.0, vzdálenost vycházející z kosinové podobnosti

6.2.2 Experimenty s vektorovým modelem, kde každý atribut představuje počet příspěvků daného uživatele odeslaného v daném časovém rozmezí

Ve většině případu byly dosaženy nejlepší výsledky při shlukování do dvou shluků a to u všech použitých indexů pro validaci shluků, viz. tabulky 20 a 21. Pouze u xie - beni

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	9	1	-	-	-	-	-	-	-	-
4	5	1	9	4	-	-	-	-	-	-
8	3	1	3	5	6	1	1	1	-	-
10	1	1	3	9	2	7	3	14	1	1
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	91	99	-	-	-	-	-	-	-	-
4	85	80	90	83	-	-	-	-	-	-
8	68	71	78	84	82	79	78	72	-	-
10	19	15	4	36	14	54	9	42	6	21

Tabulka 18: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., fuzziness koeficient - 1.2

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	1	-	-	-	-	-	-	-	-
4	1	0	0	0	-	-	-	-	-	-
8	1	1	0	0	0	0	0	0	-	-
10	1	0	0	1	0	0	0	0	0	0
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	99	99	-	-	-	-	-	-	-	-
4	99	99	99	98	-	-	-	-	-	-
8	99	99	99	98	99	98	99	99	-	-
10	29	25	33	94	25	97	26	24	35	21

Tabulka 19: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., fuzziness koeficient - 2.0

indexu při využití vzdálenosti vycházející z kosinové podobnosti pro fuzziness koeficient 2.0 dochází k lepším výsledkům u shlukování do osmi shluků.

Z obrázku 18 a tabulky 22 je patrné, že oproti předcházejícímu vektorovému modelu, u tohoto modelu dochází k mírnému překryvu již při nejnižším možném fuzziness

Euklidovská míra, fuzziness koeficient - 1.2			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.99**	**0.001**	**0.01**
4	**0.99**	0.0017	0.02
8	**0.99**	0.0026	0.04
10	**0.99**	0.0005	0,06
Míra vycházející z kosinové podobnosti, fuzziness koeficient - 1.2			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.85**	**0.36**	**1.56**
4	0.76	0.66	2.2
8	0.75	0.77	4.15
10	0.79	0.65	2.44

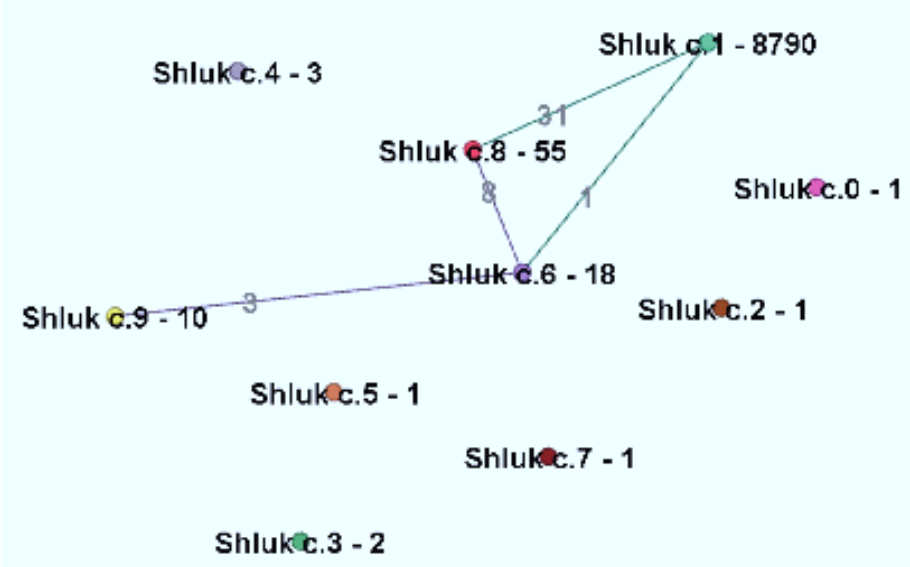
Tabulka 20: Fuzzy c-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**

Euklidovská míra, fuzziness koeficient - 2.0			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	0.997	**0.007**	**0.011**
4	**0.998**	0.035	0.036
8	0.84	0.4	0.28
10	0.67	0.79	0.6
Míra vycházející z kosinové podobnosti, fuzziness koeficient - 2.0			
Počet shluků	**Partition koeficient**	**Partition entropy**	**Xie - Beni**
2	**0.52**	**0.97**	4.61
4	0.257	1.98	153808.8
8	0.15	2.88	547.85
10	0.13	3.14	**3.41**

Tabulka 21: Fuzzy c-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**

koeficientu, což může být dáno nižším počtem atributů objektu. U vyššího fuzziness koeficientu, obr. 19 a tab. 23, dochází k většímu překryvu u shlukování do většího počtu shluků.

Při použití vzdálenosti vycházející z kosinové podobnosti je patrné, že při využiti tohoto vektorového modelu vlivem menšího počtu atributu objektů, 24 oproti až 30000 možných atributů předcházejícího vektorového modelu dochází k rovnoměrnějšímu rozložení objektů do shluků. U menšího fuzziness koeficientu 1.2 je počet objektů, které patří výhradně do daného shluku výrazně větší než u vyššího fuzziness koeficientu. K překryvům dochází u této míry vzdálenosti při obou použitých fuzziness koeficientech.

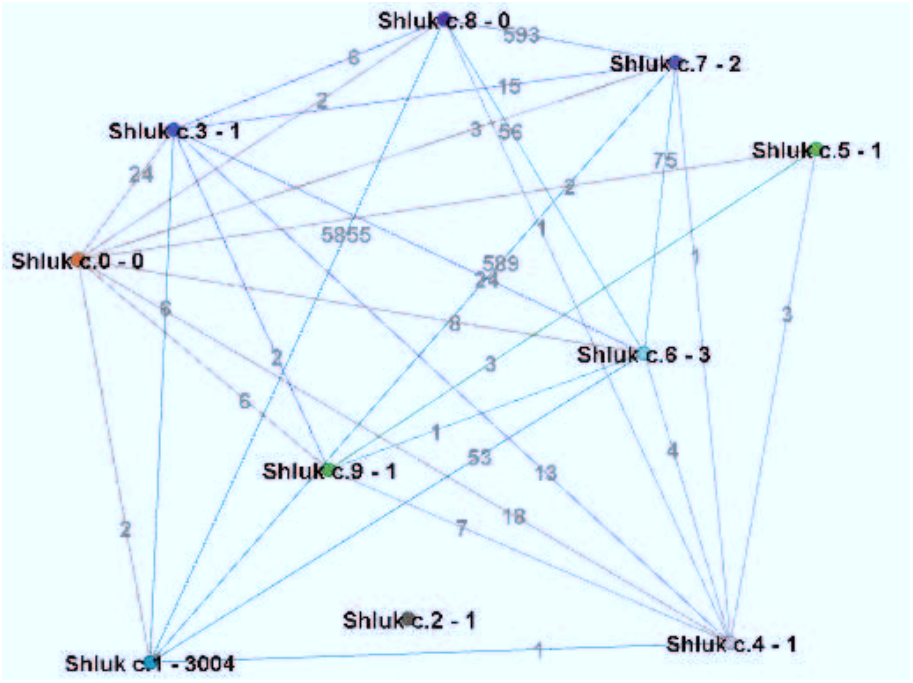


Obrázek 18: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, 10 shluků, prahová hodnota - 0.1, fuzziness - 1.2, euklidovská vzdálenost

	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	8895	19	-	-	-	-	-	-	-	-
4	8	8871	1	31	-	-	-	-	-	-
8	1	8826	1	1	17	4	51	2	-	-
10	1	8790	1	2	3	1	18	1	55	10
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	8904	28	-	-	-	-	-	-	-	-
4	12	8880	1	43	-	-	-	-	-	-
8	1	8841	1	1	22	6	69	2	-	-
10	1	8821	1	2	3	1	29	1	93	13

Tabulka 22: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a fuzziness koef. 1.2

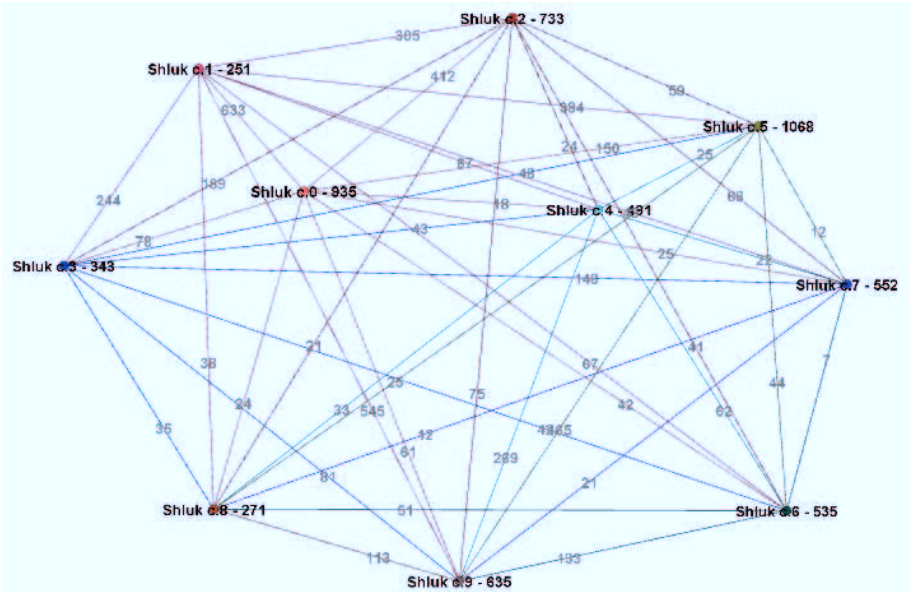
U vyššího fuzziness koeficientu dochází k přiřazení objektů do většího počtu shluku než u nižšího. Viz obr. 20 a 21 a tabulky 24 a 25.



Obrázek 19: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, 10 shluků, prahová hodnota - 0.1, fuzziness - 2.0, euklidovská vzdálenost

	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	0	8872	-	-	-	-	-	-	-	-
4	4	8761	1	4	-	-	-	-	-	-
8	0	6324	1	1	1	1	6	0	-	-
10	1	3004	1	1	1	1	3	2	0	1
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	46	8918	-	-	-	-	-	-	-	-
4	37	8906	11	154	-	-	-	-	-	-
8	30	8874	1	47	21	8	147	2558	-	-
10	29	8859	1	41	20	4	87	614	5859	8

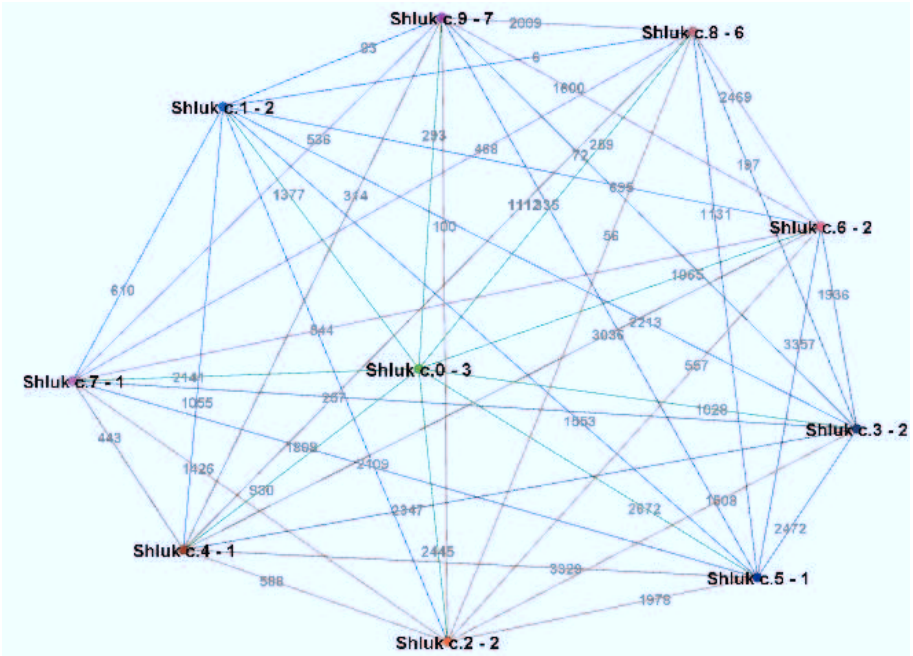
Tabulka 23: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a fuzziness koef. 2.0



Obrázek 20: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, 10 shluků, prahová hodnota - 0.1, fuzziness - 1.2, vzdálenost vycházející z kosinové podobnosti

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	2065	3980	-	-	-	-	-	-	-	-
4	1307	1780	1266	999	-	-	-	-	-	-
8	1143	227	138	651	1095	875	1046	669	-	-
10	935	251	733	343	491	1068	535	552	271	635
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 1.2									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	4943	6858	-	-	-	-	-	-	-	-
4	3747	3595	3121	3448	-	-	-	-	-	-
8	2183	1859	1451	1499	2131	1632	2058	1092	-	-
10	2054	1849	1602	947	934	1648	875	806	506	1694

Tabulka 24: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., fuzziness koeficient - 1.2



Obrázek 21: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, 10 shluků, prahová hodnota - 0.1, fuzziness - 2.0, vzdálenost vycházející z kosinové podobnosti

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	9	7	-	-	-	-	-	-	-	-
4	10	0	0	0	-	-	-	-	-	-
8	20	0	1	0	4	0	8	4	-	-
10	3	2	2	2	1	1	2	1	6	7
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, fuzziness koeficient - 2.0									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	8916	8914	-	-	-	-	-	-	-	-
4	8888	8912	8913	8912	-	-	-	-	-	-
8	4774	7922	8470	7769	6522	8497	3569	5432	-	-
10	3445	2999	3227	3596	3733	5257	4815	2472	3116	2357

Tabulka 25: FCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaného v daném časovém rozmezí**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., fuzziness koeficient - 2.0

6.3 Experimenty s Rough C-means

Při použití Rough C-means pozorujeme výsledky při změně nastavení metrik a hraniční hodnoty wlow - 0.5 a 0.9, tzn. že musíme při tomto výpočtu příslušnosti objektů ke shlukům pracovat s normovanými hodnotami.

6.3.1 Vektorový model, kde každý atribut představuje počet příspěvků daného uživatele v dané diskuzi

Pro obě hodnoty dolní aproximace wlow i pro obě použité metriky bylo nejlepších výsledků dosaženo pro shlukování do 2 shluků, viz tab. 26 a 27. Pro euklidovskou metriku při změně hodnoty dolní aproximace nedochází ke změně hodnoty validačního indexu.

Euklidovská míra, wlow - 0.5	
Počet shluků	**Davies - Bouldin index**
2	**0.045**
4	0.08
8	0.061
10	0.065
Míra vycházející z kosinové podobnosti, wlow - 0.5	
Počet shluků	**Davies - Bouldin index**
2	**1.53**
4	3.2
8	5.4
10	5.22

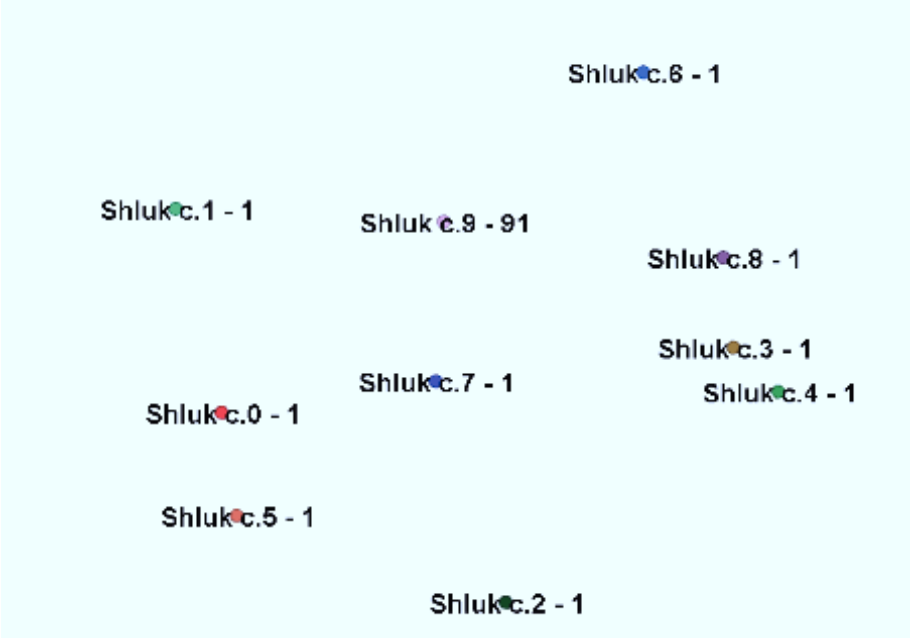
Tabulka 26: Rough C-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**

Z obr. 22 a 23 a tabulek 28 a 29 při využití euklidovské metriky je patrné, že dochází ke stejným výsledkům, tj. rozdělení objektů do shluků, jak při použití hodnoty dolní aproximace 0.5, tak i při použití hodnoty dolní aproximace 0.9. Zároveň nedochází k překryvům.

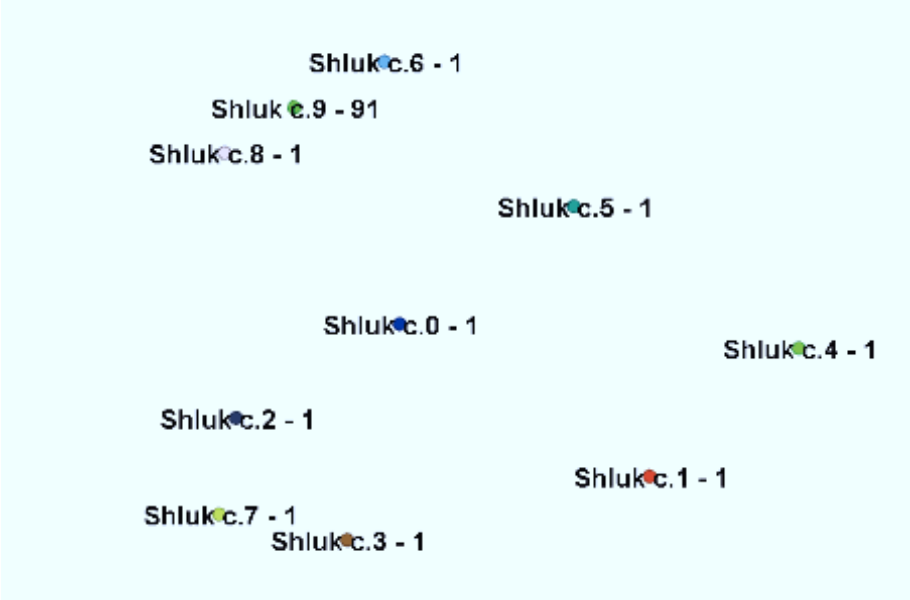
U metriky vycházející z kosinové podobnosti, viz obr. 24 a 25 a tab. 30 a 31 dochází k o něco větším překryvům mezi shluky při nastavení vyšší hodnoty dolní aproximace (0.9). Při využití této metriky, při obou použitých hodnotách dolní aproximace, málo objektů patří pouze k jednomu ze shluků.

Euklidovská míra, wlow - 0.9	
Počet shluků	**Davies - Bouldin index**
2	**0.045**
4	0.08
8	0.061
10	0.065
Míra vycházející z kosinové podobnosti, wlow - 0.9	
Počet shluků	**Davies - Bouldin index**
2	**0.23**
4	0.615
8	0.65
10	0.64

Tabulka 27: Rough C-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**



Obrázek 22: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, wlow - 0.5, euklidovská vzdálenost



Obrázek 23: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, wlow - 0.9, euklidovská vzdálenost

	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	99	-	-	-	-	-	-	-	-
4	1	1	1	97	-	-	-	-	-	-
8	1	1	1	1	1	1	1	93	-	-
10	1	1	1	1	1	1	1	1	1	91
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	99	-	-	-	-	-	-	-	-
4	1	1	1	97	-	-	-	-	-	-
8	1	1	1	1	1	1	1	93	1	-
10	1	1	1	1	1	1	1	1	1	91

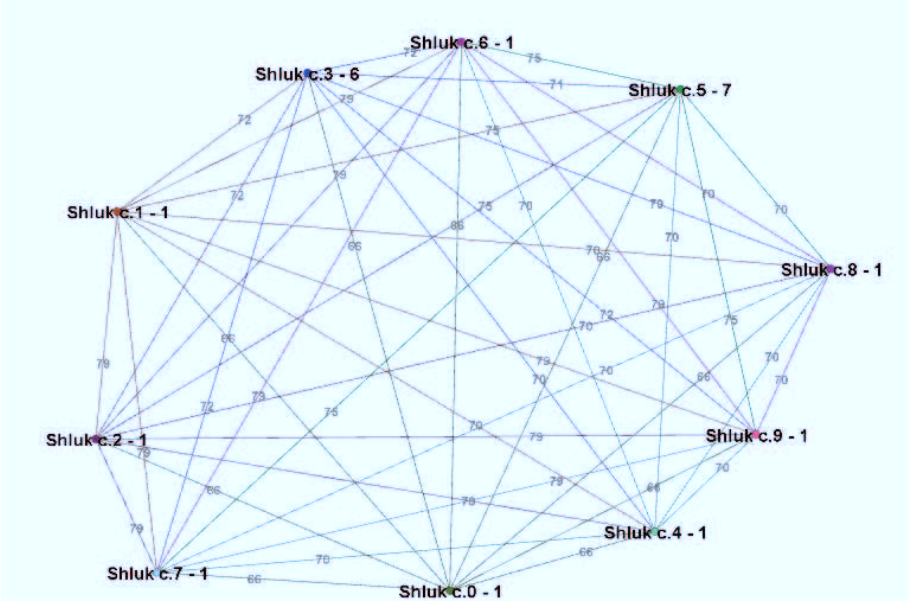
Tabulka 28: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a wlow 0.5

6.3.2 Vektorový model, kde každý atribut představuje počet příspěvků daného uživatele odeslaného v daném časovém rozmezí

Stejně jako u předchozího vektorového modelu i zde bylo dosaženo při využití obou použitých typů vzdálenosti a obou hodnot dolních aproximací stejného výsledku. Nejlepšího výsledku shlukování bylo dosaženo při shlukování do dvou shluků, viz tabulky 32 a 33.

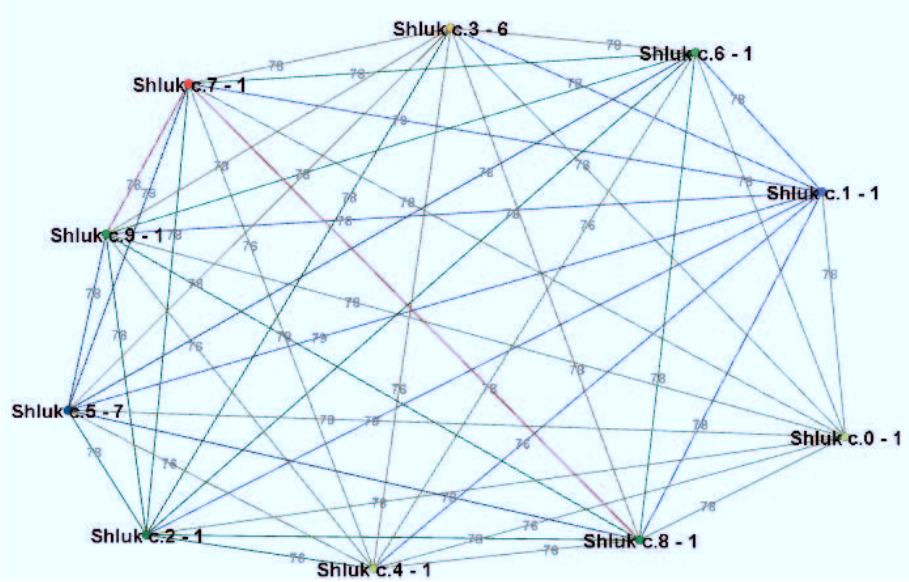
	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	99	-	-	-	-	-	-	-	-
4	1	1	1	97	-	-	-	-	-	-
8	1	1	1	1	1	1	1	93	1	-
10	1	1	1	1	1	1	1	1	1	91
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	99	-	-	-	-	-	-	-	-
4	1	1	1	97	-	-	-	-	-	-
8	1	1	1	1	1	1	1	93	1	-
10	1	1	1	1	1	1	1	1	1	91

Tabulka 29: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a wlow 0.9



Obrázek 24: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, wlow - 0.5, vzdálenost vycházející z kosinové podobnosti

Z tabulek 34 a 35 je patrné, že u euklidovské metriky při využití menší hodnoty dolní aproximace wlow při shlukování do 2 a 4 shluků, patří téměř všechny objekty přímo do



Obrázek 25: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, 10 shluků, prahová hodnota - 0.1, wlow - 0.9, vzdálenost vycházející z kosinové podobnosti

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	14	-	-	-	-	-	-	-	-
4	1	1	1	6	-	-	-	-	-	-
8	1	1	1	6	1	7	1	1	-	-
10	1	1	1	6	1	7	1	1	1	1
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	86	99	-	-	-	-	-	-	-	-
4	78	92	92	89	-	-	-	-	-	-
8	69	82	82	80	73	84	82	82	-	-
10	67	80	80	78	71	82	80	80	71	80

Tabulka 30: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., wlow - 0.5

některého ze shluků. Při shlukování do 8 a 10 shluků patří většina objektů do více shluků.

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	1	-	-	-	-	-	-	-	-
4	1	1	1	6	-	-	-	-	-	-
8	1	1	1	6	1	7	1	1	-	-
10	1	1	1	6	1	7	1	1	1	1
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	99	99	-	-	-	-	-	-	-	-
4	92	92	92	97	-	-	-	-	-	-
8	81	82	81	86	79	88	81	82	-	-
10	79	80	79	84	77	86	79	80	79	79

Tabulka 31: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v dané diskuzi**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., wlow - 0.9

Euklidovská míra, wlow - 0.5	
Počet shluků	**Davies - Bouldin index**
2	**0.003**
4	0.91
8	0.88
10	96.88
Míra vycházející z kosinové podobnosti, wlow - 0.5	
Počet shluků	**Davies - Bouldin index**
2	**12.86**
4	14.24
8	41.21
10	55.6

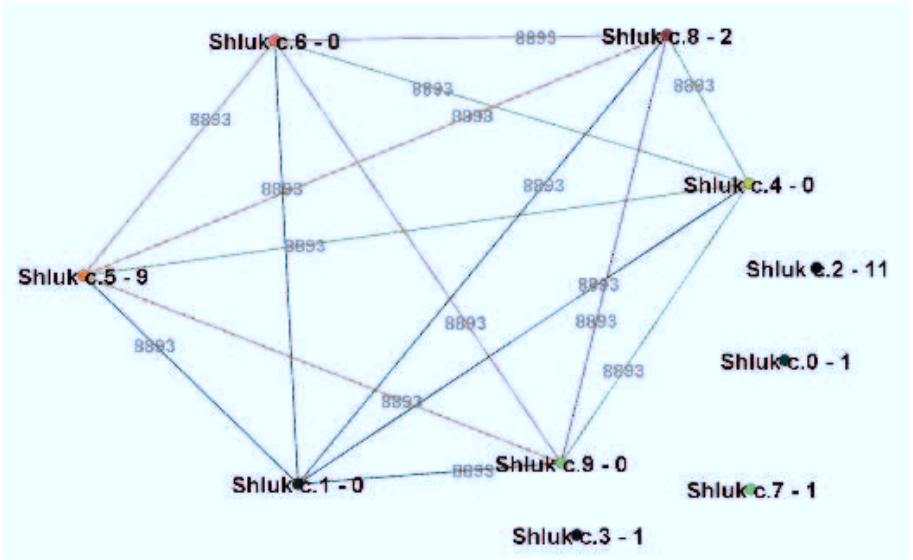
Tabulka 32: Rough C-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**

U vyšší hodnoty dolní aproximace patří téměř všechny objekty výhradně k některému ze shluků. Minimum objektů patří do více shluků. Viz tabulky 34 a 35.

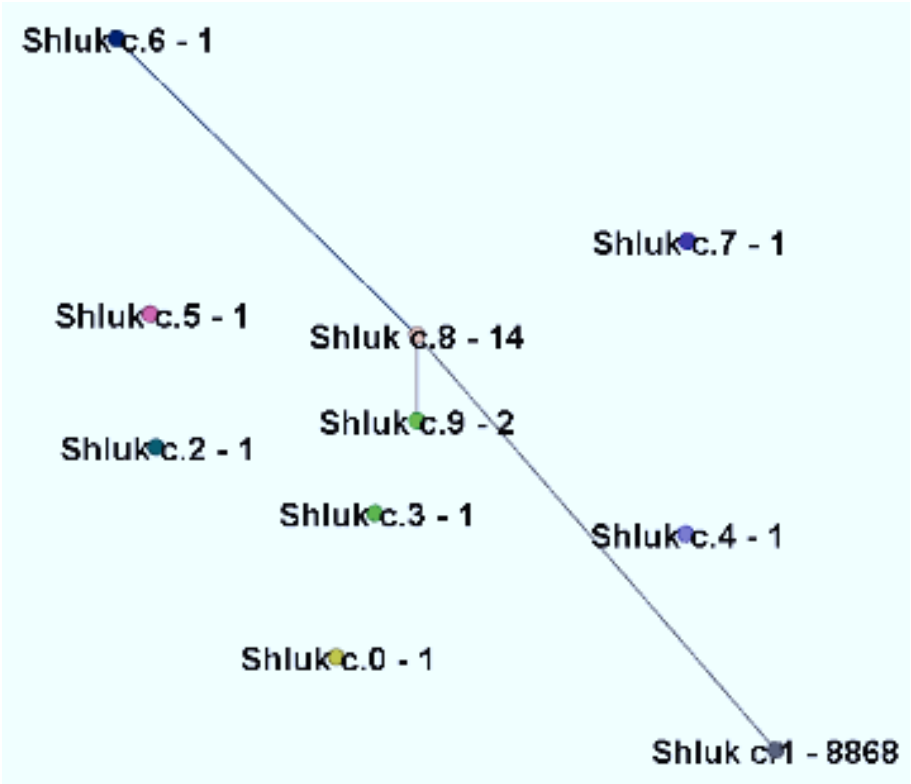
U druhé použité metriky vycházející z kosinové podobnosti při niž dochází k lepšímu rozdělení objektů do shluků, je při nižší hodnotě dolní aproximace počet objektů, které výhradně patří pouze do jednoho ze shluků menší než při vyšší hodnotě dolní aproximace. Počet objektů patřících do více shluků je naopak u nižší hodnoty dolní aproximace větší než u vyšší hodnoty dolní aproximace. Viz obr. 28 a 29 a tabulky 36 a 37.

Euklidovská míra, wlow - 0.9	
Počet shluků	**Davies - Bouldin index**
2	**0.003**
4	0.68
8	0.45
10	0.405
Míra vycházející z kosinové podobnosti, wlow - 0.9	
Počet shluků	**Davies - Bouldin index**
2	**2.87**
4	3.2
8	2.43
10	2.15

Tabulka 33: Rough C-means, validace shlukování pro vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**



Obrázek 26: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, 10 shluků, prahová hodnota - 0.1, wlow - 0.5, euklidovská vzdálenost



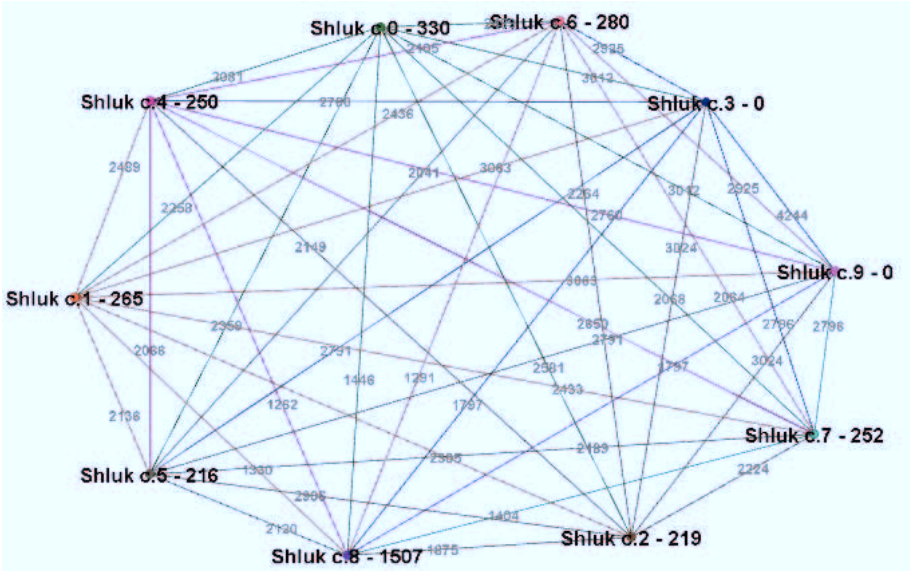
Obrázek 27: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, 10 shluků, prahová hodnota - 0.1, wlow - 0.9, euklidovská vzdálenost

	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	8922	-	-	-	-	-	-	-	-
4	1	8874	19	3	-	-	-	-	-	-
8	1	0	1	1	7	1	15	2	-	-
10	1	0	11	1	0	9	0	1	2	0
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	8922	-	-	-	-	-	-	-	-
4	1	8900	45	3	-	-	-	-	-	-
8	1	8893	1	1	9	3	8908	2	-	-
10	1	8893	11	1	8893	8907	8893	1	8900	8893

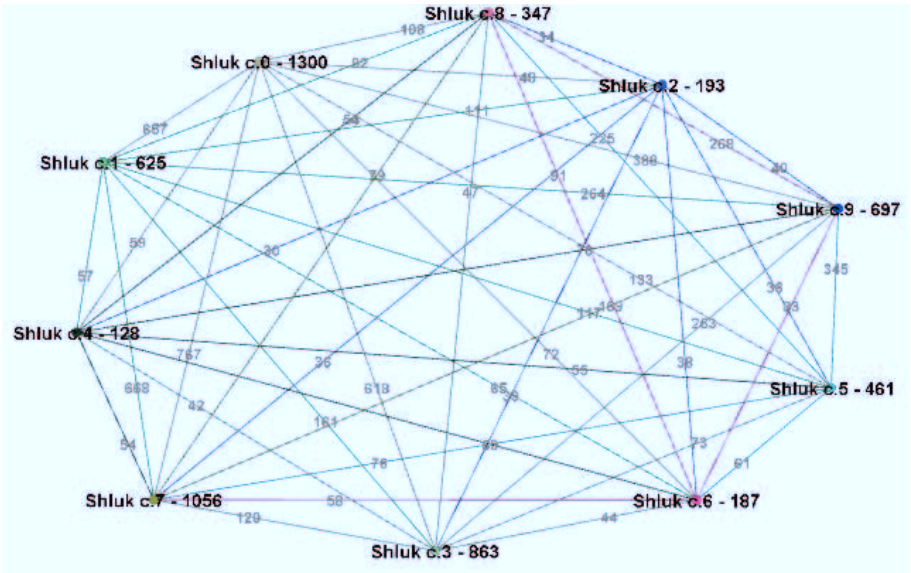
Tabulka 34: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a wlow 0.5

	Počet objektu patřící pouze do daného shluku. **Euklidovská vzdálenost**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	8922	-	-	-	-	-	-	-	-
4	1	8891	23	3	-	-	-	-	-	-
8	1	8874	1	1	1	1	19	2	-	-
10	1	8868	1	1	1	1	1	1	14	2
	Počet objektu patřící do daného shluku (i do jiných). **Euklidovská vzdálenost**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	1	8922	-	-	-	-	-	-	-	-
4	1	8896	28	3	-	-	-	-	-	-
8	1	8895	1	1	3	3	40	2	-	-
10	1	8895	1	1	1	1	3	1	46	5

Tabulka 35: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, počet objektů patřících do jednotlivých shluků pro euklidovskou vzdálenost a wlow 0.9



Obrázek 28: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, 10 shluků, prahová hodnota - 0.1, wlow - 0.5, vzdálenost vycházející z kosinové podobnosti



Obrázek 29: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, 10 shluků, prahová hodnota - 0.1, wlow - 0.9, vzdálenost vycházející z kosinové podobnosti

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	2062	2280	-	-	-	-	-	-	-	-
4	665	1619	870	708	-	-	-	-	-	-
8	468	0	1308	366	247	0	294	601	-	-
10	330	265	219	0	250	216	280	252	1507	0
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, wlow - 0.5									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	6643	6861	-	-	-	-	-	-	-	-
4	4342	4928	4303	4501	-	-	-	-	-	-
8	3876	4490	4195	3983	4164	4490	4083	3794	-	-
10	3561	3546	3704	4244	3320	3713	3414	3312	4041	4244

Tabulka 36: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, Počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., wlow - 0.5

	Počet objektu patřící pouze do daného shluku. **Vzdálenost vycházející z kos. podobn.**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	2397	4750	-	-	-	-	-	-	-	-
4	1766	1369	1014	1846	-	-	-	-	-	-
8	1351	718	200	867	503	491	735	1109	-	-
10	1300	625	193	863	128	461	187	1056	347	697
	Počet objektu patřící do daného shluku (i do jiných). **Vzdálenost vycházející z kos. podobn.**, wlow - 0.9									
k shluků	c_0	c_1	c_2	c_3	c_4	c_5	c_6	c_7	c_8	c_9
2	4173	6526	-	-	-	-	-	-	-	-
4	3857	3160	2816	3157	-	-	-	-	-	-
8	1234	2527	2358	561	927	1266	1924	2452	-	-
10	3091	1837	337	1723	280	973	416	2267	794	1674

Tabulka 37: RCM, vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele v daném časovém období**, počet objektů patřících do jednotlivých shluků pro vzdálenost vycházející z kos. podobn., wlow - 0.9

7 Závěr

Cílem této diplomové práce bylo provést přehled soft neboli překrývajících se shlukovacích algoritmů, implementaci některých z nich a provést jejich aplikaci nad vybranou datovou kolekcí.

Nejdříve jsme provedli extrakci dat z vybraného diskuzního fóra, v tomto případě matematické diskuzní fórum, jež nám následně sloužila jako reálná kolekce dat. Z takto extrahovaných dat, uložených v databázi jsme vytvořili tři vektorové modely, a sice vektorový model, kde u každého objektu je pouze jeden atribut, a **to průměrný čas, ve kterém daný uživatel vkládá do diskuzí své příspěvky**. Vektorový model, kde každý atribut představuje **počet příspěvků daného uživatele odeslaných v daném časovém rozmezí**. Vektorový model, kde každý atribut představuje **počet příspěvku daného uživatele v dané diskuzi**. U druhého a třetího vektorového modelu jsme pak vzhledem k tomu, že obsahují velké množství nulových hodnot provedli úpravu tak, aby vektorový model tyto nulové hodnoty neobsahoval.

Tyto vektorové modely jsme následně použili při práci s vybranými shlukovacími algoritmy. Implementovány byly algoritmy k-means, fuzzy c-means a rough c-means, u níž jsme měnili nastavení vstupních parametrů a zkoumali, jak se projeví výsledky shlukování při těchto změných.

Při aplikaci vstupních dat jsme zjistili nevhodnost použití posledního vektorového modelu, kde každý atribut představuje **počet příspěvku daného uživatele v dané diskuzi** a to kvůli tomu, že v tomto vektorovém modelu existuje až přibližně 30000 atributů, což je velmi časově náročné při výpočtech, kdy musíme porovnávat zda mají objekty společné atributy a to při odstranění nulových hodnot. Tento vektorový model není vhodný pro použití žádné z uvedených metrik vzdálenosti.

Dále pak z výsledku rozdělení objektů do shluků jsme určili nevhodnost použití euklidovské metriky. Při volbě počátečních center metodou KKZ, kdy jsou jako centra zvoleny nejvzdálenější objekty dojde k přiřazení většiny objektů k centru s nejmenším počtem atributů a nejmenší hodnotou těchto atributů a to i v případě, že dojde k přepočtu nových center. Z tohoto důvodu je pro práci s těmito daty použít vzdálenost vycházející z kosinové, popř. jaccardovy podobnosti.

Při aplikaci algoritmů na dané vektorové modely jsme si ověřili, že je dobré znát zpracovávaná data z důvodu volby prahové hodnoty. U algoritmu k-means nám tato prahová hodnota určuje, která data budou uložena do souboru pro vizualizační nástroj Gephi. Pokud bychom zvolili příliš vysokou hodnotu, výstupní soubor by byl příliš velký a pro Gephi těžko zpracovatelný. Vysoká prahová hodnota by pak zase znamenala velký počet odlehlých hodnot, takže výsledný graf by byl značně nesouvislý. Volené prahové hodnoty pro dané míry vzdálenosti jsou uvedeny u jednotlivých vizualizovaných grafů v kapitole experimenty.

U algoritmu fuzzy c-means jsme se přesvědčili, že míra překryvu jednotlivých shluků záleží na volbě fuzziness koeficientu, kdy nižší fuzziness koeficient znamená menší míru překryvu než koeficient větší. Dále také závisí na zvolené prahové hodnotě, tedy pokud bude příslušnost objektů větší než zadaná prahová hodnota, bude patřit objekt k danému shluku.

Při aplikaci algoritmu rough c-means jsme si ověřili, že při při nižší hodnotě dolní aproximace wlow je příslušnost objektů pouze k jednomu z daných shluků menší než u vyšší hodnoty dolní aproximace, počet objektů patřících do více shluků je při nižší hodnotě wlow větší. U vyšší hodnoty dolní aproximace je tomu naopak, počet objektů patřících pouze do jednoho ze shluků je větší a počet objektů patřících do více shluků je menší.

U všech algoritmů je rozdělení objektů do shluku lepší a rovnoměrnější pro vzdálenost vycházející z kosinové podobnosti. Pro naše testovaná data bylo nejlepších výsledků shlukování většinou dosaženo při shlukování do dvou shluků.

8 Reference

[1] MELOUN, Milan; MILITKÝ, Jiří. *Přednosti analýzy shluků ve vícerozměrné statistické analýze*. Sborník přednášek z konference: Zajištění kvality analytických výsledků, str. 29-46, Medlov, 22. - 24. 3. 2004, ISBN 80-86380-22-X

[2] ADAM, Martin. *Knihovna pro práci s řídkými maticemi*[online]. Brno : Masarykova univerzita, 2010. 30 s. Bakalářská práce. Masarykova univerzita, Fakulta informatiky. Dostupné z WWW: http://is.muni.cz/th/255922/fi_b/BCthesis.pdf

[3] DOLENSKÝ, Petr. *Vytvoření a analýza sociální sítě nad vybranými zdroji na Webu*[online]. Ostrava : Vysoká škola báňská - Technická univerzita Ostrava, 2010. 34 s. Bakalářská práce. Vysoká škola báňská - Technická univerzita Ostrava. Dostupné z WWW: https://dspace.vsb.cz/bitstream/handle/10084/78854/DOL199_FEI_B2646 _2612R025 _2010.pdf?sequence=1

[4] CLEUZIOU, Guillaume; MARTIN, Lionel; VRAIN, Christel. *PoBOC: an Overlapping Clustering Algorithm*. [online]. [cit. 2011-09-05]. Dostupný z WWW: http://www.frontiersinai.com/ecai/ecai2004/ecai04/pdf/p0440.pdf.

[5] KELBEL, Jan; ŠILHÁN, David. *Shluková analýza*. [online]. [cit. 2011-09-16]. Dostupný z WWW: http://staff.utia.cas.cz/nagy/skola/Projekty/Classification/ShlukovaAnalyza.pdf

[6] LAMPINEN, Timo; KOIVISTO, Hannu; HONKANEN, Tapani. *Profiling network applications with fuzzy c-means clustering anf self-organizing map*.[online]. [cit. 2011-09-05]. Tampere University of Technolog. Dostupný z WWW: http://ae.tut.fi/research/AIN/Publications/FSKD2002Lampinen.pdf

[7] PAWLAK, Zdislaw. *Rough sets*. [online]. [cit. 2011-09-05]. Gliwice, Poland : Institute of Theoretical and Applied Informatics, Polish Academy of Sciences. Dostupné z WWW: http://chc60.fgcu.edu/images/articles/RoughSetsRep29.pdf

[8] JIANBIN, Chen; FANG, Deying; TONG, Shi. A. *Graph Partition-based Soft Clustering Algorithm*. Intelligent Information Technology Application. [online]. 2008. [cit. 2011-09-05]. Dostupný z WWW: http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4739829

[9] YEN-LIANG, Chen; HUI-LING, Hu. *An overlapping cluster algorithm to provide non-exhaustive clustering*. European Journal of Operational Research. [online]. 2006. [cit. 2011-09-05]. Dostupný z WWW: http://www.sciencedirect.com/science/article/pii/S0377221705006727

[10] Wikipedia: http://en.wikipedia.org/wiki/Vector_space_model [online]. [cit. 2011-11-15]. San Francisco (CA): Wikimedia Foundation, 2005-12-08.

[11] HOUDEK, Petr, Josef SCHWARZ a Václav SNÁŠEL. *Moderní metody vyhledávání dokumentů v rozsáhlých plnotextových databázích: Příklad vektorového modelu*. [online]. [cit. 2011-09-05]. Dostupné z: http://www.ikaros.cz/dokumenty/amphor.pdf

[12] ABRAHAM, Ajith, Rafael FALCÓN a Rafael BELLO. Rough Set Theory: A true landmark in data analysis. Berlin: Springer, 2009. ISBN 978-3-540-899.

[13] ZEXUAN, Ji, Sun QUANSEN, YONG, Chen QIANG, Xia DESHEN a Feng DAGAN. *Generalized rough fuzzy c-means algorithm for brain MR image segmentation*. [online]. [cit. 2012-04-11]. Dostupné z: http://www.sciencedirect.com/science/article/pii/S0169260711002896

[14] BRUN, Marcel, Chao SIMA, Jianping HUA, James LOWEY, Brent CARROLL, Edward SUH a R. DOUGHERTY. *Model-based evaluation of clustering validation measures*. [online]. [cit. 2012-04-11]. Dostupné z: http://www.sciencedirect.com/science/article/pii/S0031320306003104

[15] RENDÓN, Eréndira, Itzel M. ABUNDEZ, Citlalih GUTIERREZ, Sergio DIÁZ, Alejandra ARIZMENDI, M. QUIROZ a Elsa H.ARZATE. *A comparison of internal and external cluster validation indexe*. [online]. [cit. 2012-04-11]. Dostupné z: http://www.wseas.us/e-library/conferences/2011/Mexico/CEMATH/CEMATH-26.pdf

[16] SHU-ZHONG, Yang a Lu SI-WEI. *A novel algorithm for initializing clustering centers* [online]. 2005. [cit. 2011-10-12]. Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01527930

[17] K-Means Clustering [online]. [cit. 2011-11-10]. Dostupné z: http://home.dei.polimi.it/matteucc/ Clustering/tutorial_html/kmeans.html

[18] SUSHMITA, Mitra, Haider BANKA a Witold PEDRYCZ,. *Rough–Fuzzy Collaborative Clustering* [online]. 2006. [cit. 2012-04-21]. Dostupné z: http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=01658293&tag=1

[19] KUO-LUNG, Wu a Yang MIIN-SHEN. *A cluster validity index for fuzzy clustering*. [online]. 2004. [cit. 2012-04-11] Dostupné z: http://www2.math.cycu.edu.tw/TEACHER/MSYANG/fuzzy-e/yang-pdf/yang-n-28-cluster-validity.pdf

[20] Jaccard Index. [online]. [cit. 2012-02-12]. Dostupné z: http://www.enotes.com/topic/Jaccard_index

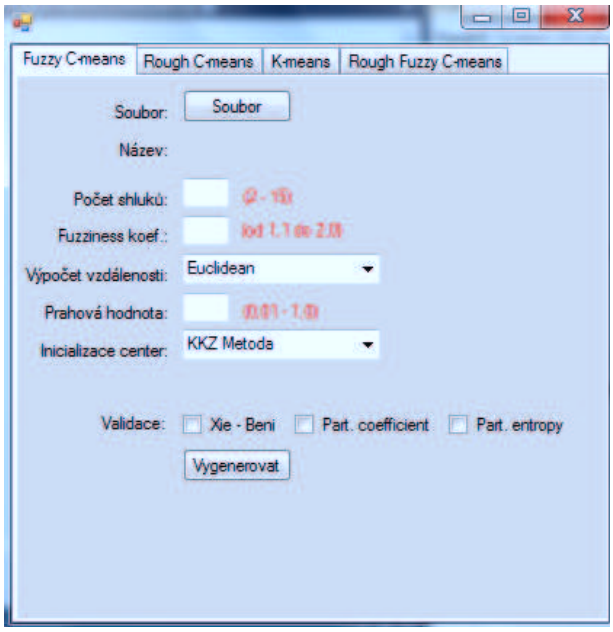
A Manuál a požadavky pro spuštění

Požadavky na spuštění

- OS Microsoft Windows XP, Vista, 7
- .NET framework 4.0

Aplikace je vytvořena tak, že stačí zadat parametry do uživatelského rozhraní pro daný shlukovací algoritmus a provést generování souborů, viz 30. Na dalším obrázku 32 je ukázka vyplněných parametrů. Důležité je, aby v desetinných číslech nebyla tečka, ale **čárka**.

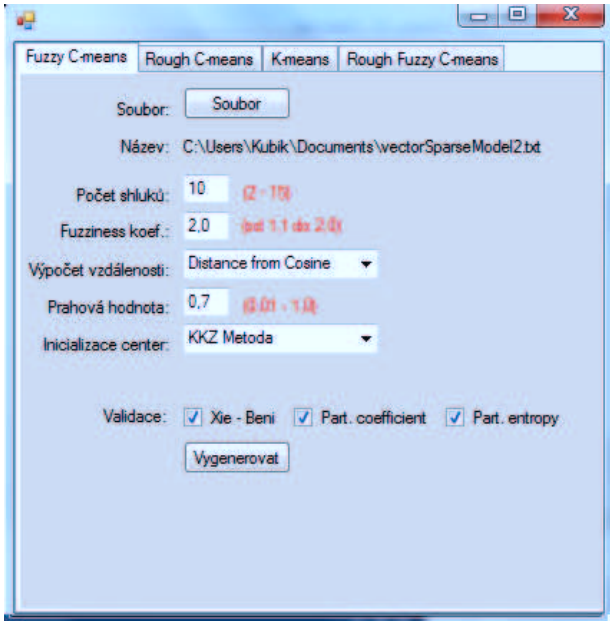
Na obrázcích 34 a 35 je ukázka výstupních souborů.



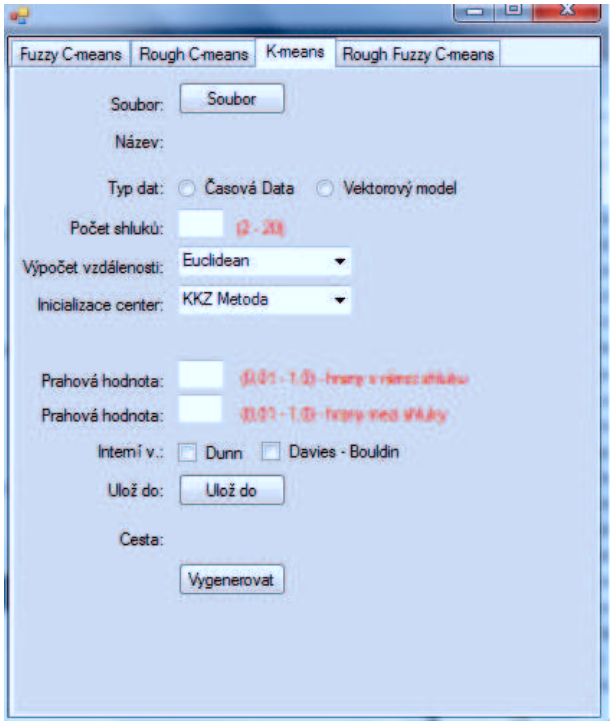
Obrázek 30: Aplikace

Obrázek 31: Okno aplikace.

Na obrázku 33 je ukázka panelu pro algoritmus K-means. Jestliže chceme pracovat s vektorovým modelem, kde má každý objekt jen jeden atribut, a to průměrný čas odesílání příspěvku (soubor vectorSparseModelTime.txt), pak vybereme u řádku „typ dat“ časová data, v ostatních případech vektorový model.



Obrázek 32: Okno aplikace s vyplněnými vstupními parametry.



Obrázek 33: Okno aplikace pro K-means

Obrázek 34: Výstupní textový soubor.

```
graph [

 node [
  id 0 label "Shluk c.0 - 294"
graphics [  fill "#9C373A"
 ]
]
 node [
  id 1 label "Shluk c.1 - 540"
graphics [  fill "#E2CAD3"
 ]
]

.....................

edge [
  id 1
 source 0
  target 1
 label "595"  ]
edge [
  id 2
 source 0
  target 2
 label "928"  ]
```

Obrázek 35: Aplikace

Obrázek 36: Výstupní gml soubor pro Gephi.

B Obsah přiloženého DVD

DVD obsahuje následující adresáře

- \textDP
  - Text DP práce
- \zadani
  - Zadání DP práce
- \vectorModels
  - Používané vektorové modely
- \dtb
  - Databáze, ze kterých byly vytvořeny vektorové modely
- \Clustering
  - Zdrojové kódy a knihovny aplikace, včetně tříd pro extrakci dat a vytvoření vektorových modelů
- \Clustering \Clustering\bin\Debug
  - Spustitelný exe soubor Clustering.exe